

intel®

**MCS-85™  
USER'S  
MANUAL**

**September 1978**

©Intel Corporation 1973  
9800366E

**\$5.00**

**MCS  
85**

**MCS  
85**

**MCS  
85**

**MCS  
85**



# **MCS-85™ USER'S MANUAL**

**September 1978**

This manual supersedes all  
previous MCS-85 User's Manuals.



# MCS-85™ User's Manual

## Table of Contents

<b>CHAPTER 1</b>	
Introduction .....	1-1
<b>CHAPTER 2</b>	
Functional Description .....	2-1
<b>CHAPTER 3</b>	
System Operation and Interfacing .....	3-1
<b>CHAPTER 4</b>	
The Instruction Set .....	4-1
Instruction Set Index .....	4-19
<b>CHAPTER 5</b>	

### DEVICE SPECIFICATIONS

#### MCS-85

8085A 8-Bit Microprocessor .....	5-1
8155/56 RAM/IO/Counter-Timer .....	5-17
8185 RAM .....	5-31
8355 ROM/IO .....	5-35
8755A EPROM/IO .....	5-43

#### System Support Components

8205 High-Speed 1 Out of 8 Binary Decoder .....	5-53
8212 8-Bit Latch .....	5-59
8257 Programmable DMA Controller .....	5-69
8259A Programmable Interrupt Controller .....	5-87
8282/83 8-Bit Latches .....	5-105
8286/87 8-Bit Parallel Bidirectional Bus Drivers .....	5-109

#### Peripherals\*

8041/8741 Universal Peripheral Interface .....	5-113
8202 Dynamic RAM Controller .....	5-114
8251A Programmable Communication Interface .....	5-115
8253, 8253-5 Programmable Interval Timer .....	5-116
8255A, 8255A-5 Programmable Peripheral Interface .....	5-117
8271 Programmable Floppy Disk Controller .....	5-118
8273 Programmable HDLC/SDLC Controller .....	5-119
8275 Programmable CRT Controller .....	5-120
8278 Programmable Keyboard Interface .....	5-121
8279, 8279-5 Programmable Keyboard Display Interface .....	5-122
8291 GPIB Talker/Listener .....	5-123
8292 GPIB Controller .....	5-124
8294 Data Encryption Unit .....	5-125
8295 Dot Matrix Printer Controller .....	5-126

\*This section contains partial data sheets. For complete specifications refer to the Intel Peripheral Design Handbook.

# Table of Contents (Continued)

## CHAPTER 5 (Continued)

### Static RAMs

2114 1024x4 Bit Static RAM .....	5-127
M2114 1024x4 Bit Static RAM .....	5-131
2142 1024x4 Bit Static RAM .....	5-135

### ROMs/EPROMs

2332 32K (4Kx8) ROM .....	5-139
2364 64K (8Kx8) Bit ROM .....	5-143
2616 16K (2Kx8) Factory Programmable PROM .....	5-147
2716 16K (2Kx8) UV Erasable PROM .....	5-151
2732 32K (4Kx8) UV Erasable PROM .....	5-156
2758 8K (1Kx8) UV Erasable Low Power PROM .....	5-157
3604A, 3624A Family 4K (512x8) High Speed PROM .....	5-161
3605A, 3625A 4K (1Kx4) PROM .....	5-165

## CHAPTER 6

### DEVELOPMENT AIDS

Intellec® Microcomputer Development System Model 220 .....	6-1
Intellec® Microcomputer Development System Model 230 .....	6-5
ISIS-II Diskette Operating System .....	6-9
PL/M-80 High Level Programming Language Inteltec® Resident Compiler .....	6-13
ICE-85 MCS-85 In-Circuit Emulator .....	6-17
SDK-85 System Design Kit .....	6-21
Fortran-80 8080/8085 ANS Fortran 77 Inteltec® Resident Compiler .....	6-27

## APPENDIX 1

Applications of MCS-85 .....	A1-1
Section 1 Introduction .....	A1-1
Section 2 Detailed Application Examples .....	A1-8

## APPENDIX 2

Packaging Information .....	A2-1
-----------------------------	------

# CHAPTER 1

## Introduction

**MCS  
85**  
**MCS  
85**  
**MCS  
85**



## CHAPTER 1 INTRODUCTION TO MCS-85™

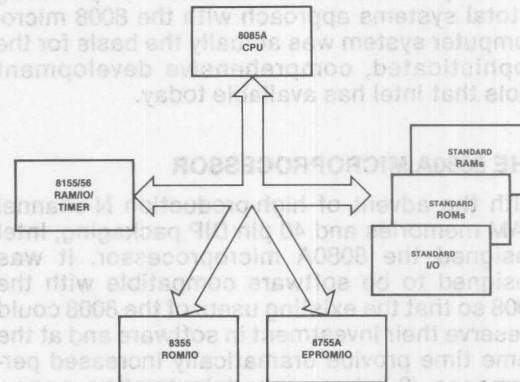
### THE MCS-85™ MICROCOMPUTER SYSTEM

The basic philosophy behind the MCS-85 microcomputer system is one of logical, evolutionary advance in technology without the waste of discarding existing investments in hardware and software. The MCS-85 provides the existing 8080 user with an increase in performance, a decrease in the component count, operation from a single 5-Volt power supply, and still preserves 100% of his existing software investment. For the new microcomputer user, the MCS-85 represents the refinement of the most popular microcomputer in the industry, the Intel 8080, along with a wealth of supporting software, documentation and peripheral components to speed the cycle from prototype to production. The same development tools that Intel has produced to support the 8080 microcomputer system can be used for the MCS-85, and additional add-on features are available to optimize system development for MCS-85.

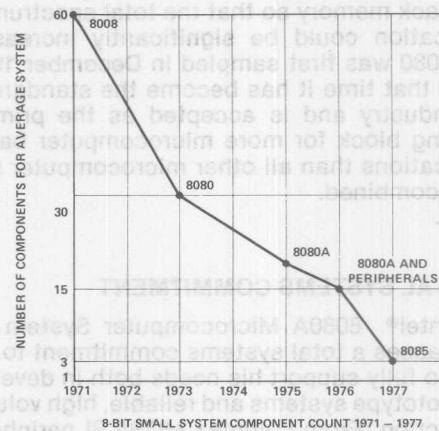
This section of the MCS-85 User's Manual will briefly detail the basic differences between the MCS-85 and MCS-80™ families. It will illustrate both the hardware and software compatibilities and also reveal some of the engineering trade-offs that were met during the design of the MCS-85. More detailed discussion of the MCS-85 bus operation and component specifications are available in Chapters: 2, 3, 4, and 5. The information provided in Chapter 1 will be helpful in understanding the basic concepts and philosophies behind the MCS-85.

### EVOLUTION

In December 1971, Intel introduced the first general purpose, 8-bit microprocessor, the 8008. It was implemented in P-channel MOS technology and was packaged in a single 18 pin, dual in-line package (DIP). The 8008 used standard semiconductor ROM and RAM and, for the most part, TTL components for I/O and general interface. It immediately found applications in byte-oriented end products such as terminals and computer peripherals where its instruction execution (20 micro-seconds), general



MCS-85™ TOTAL SYSTEM



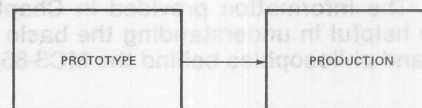
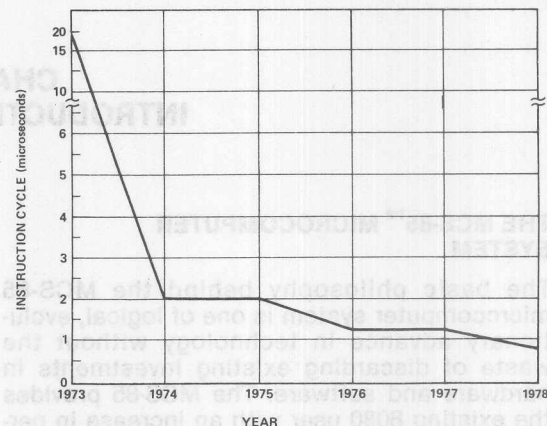
purpose organization and instruction set matched the requirements of these products. Recognizing that hardware was but a small part in the overall system picture, Intel developed both hardware and software tools for the design engineer so that the transition from prototype to production would be as simple and fast as possible. The commitment of providing a total systems approach with the 8008 microcomputer system was actually the basis for the sophisticated, comprehensive development tools that Intel has available today.

## THE 8080A MICROPROCESSOR

With the advent of high-production N-channel RAM memories and 40 pin DIP packaging, Intel designed the 8080A microprocessor. It was designed to be software compatible with the 8008 so that the existing users of the 8008 could preserve their investment in software and at the same time provide dramatically increased performance (2 micro-second instruction execution), while reducing the amount of components necessary to implement a system. Additions were made to the basic instruction set to take advantage of this increased performance and large system-type features were included on-chip such as DMA, 16-bit addressing and external stack memory so that the total spectrum of application could be significantly increased. The 8080 was first sampled in December 1973. Since that time it has become the standard of the industry and is accepted as the primary building block for more microcomputer based applications than all other microcomputer systems combined.

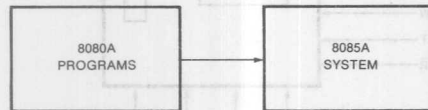
## A TOTAL SYSTEMS COMMITMENT

The Intel® 8080A Microcomputer System encompasses a total systems commitment to the user to fully support his needs both in developing prototype systems and reliable, high volume production. From complex MOS/LSI peripheral components to resident high level systems language (PL/M) the Intel® 8080 Microcomputer System provides the most comprehensive, effective solution to today's system problems.



## SOFTWARE COMPATIBILITY

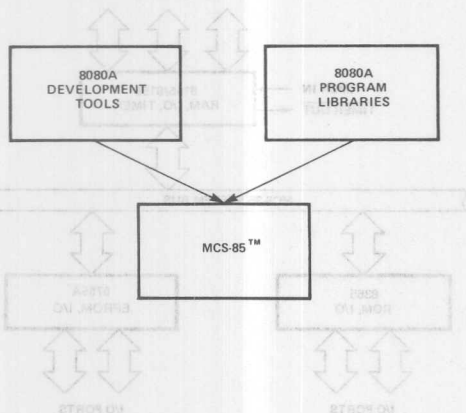
As with any computer system the cost of software development far outweighs that of hardware. A microcomputer-based system is traditionally a very cost-sensitive application and the development of software is one of the key areas where success or failure of the cost objectives is vital.



The 8085A CPU is 100% software compatible with the Intel® 8080A CPU. The compatibility is at the object or "machine code" level so that existing programs written for 8080A execution will run on the 8085A as is. The value of this becomes even more evident to the user who has mask programmed ROMs and wishes to update his system without the need for new masks.

## PROGRAMMER TRAINING

A cost which is often forgotten is that of programmer training. A new, or modified instruction set, would require programmers to relearn another set of mnemonics and greatly affect the productivity during development. The 100% compatibility of the 8085A CPU assures that no re-training effort will be required.



For the new microcomputer user, the software compatibility between the 8085A and the 8080A means that all of the software development tools that are available for the 8080A and all software libraries for 8080A will operate with the new design and thus save immeasurable cost in development and debug.

The 8085A CPU does however add two instructions to initialize and maintain hardware features of the 8085A. Two of the unused opcodes of the 8080A instruction set were designated for the addition so that 100% compatibility could be maintained.

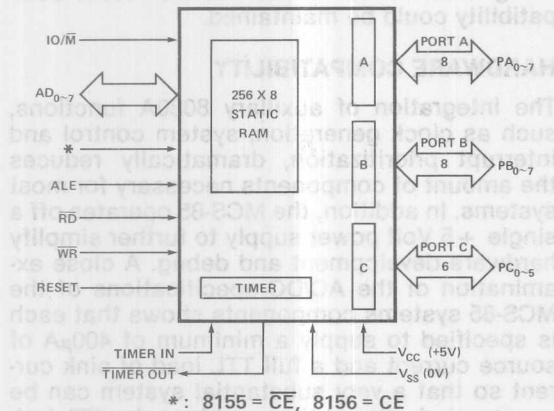
## HARDWARE COMPATIBILITY

The integration of auxiliary 8080A functions, such as clock generation, system control and interrupt prioritization, dramatically reduces the amount of components necessary for most systems. In addition, the MCS-85 operates off a single +5 Volt power supply to further simplify hardware development and debug. A close examination of the AC/DC specifications of the MCS-85 systems components shows that each is specified to supply a minimum of 400μA of source current and a full TTL load of sink current so that a very substantial system can be constructed without the need for extra TTL buffers or drivers. Input and output voltage levels are also specified so that a minimum of 350mV noise margin is provided for reliable, high-performance operation.

## PC BOARD CONSIDERATIONS

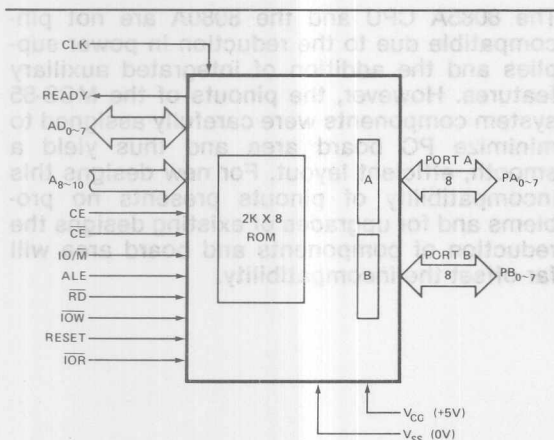
The 8085A CPU and the 8080A are not pin-compatible due to the reduction in power supplies and the addition of integrated auxiliary features. However, the pinouts of the MCS-85 system components were carefully assigned to minimize PC board area and thus yield a smooth, efficient layout. For new designs this incompatibility of pinouts presents no problems and for upgrades of existing designs the reduction of components and board area will far offset the incompatibility.

The MCS-85 was designed to minimize the amount of components required for most systems. Intel designed several new peripheral components that combine memory, I/O and timer functions to fulfill this requirement. These new peripheral devices directly interface to the multiplexed MCS-85 bus structure and provide new levels in system integration for today's designer.



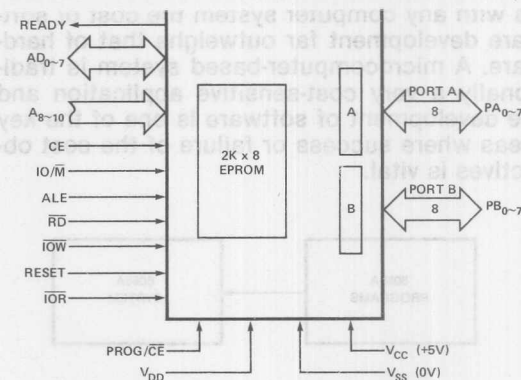
8155/8156 RAM, I/O and Timer

256 bytes RAM  
Two 8-bit ports  
One 6-bit port (programmable)  
One 14-bit programmable interval timer  
Single +5 Volt supply operation  
40 pin DIP plastic or cerdip package



8355 ROM and I/O

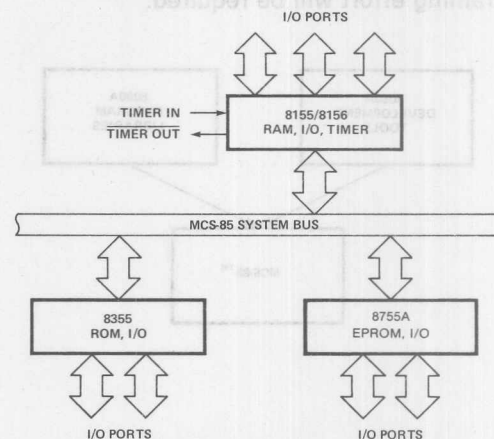
2K bytes ROM  
Two 8-bit ports (direction programmable)  
Single +5 Volt supply operation  
40 pin DIP plastic or cerdip package



8755A EPROM and I/O

Socket compatible with 8355  
2K bytes EPROM  
Two 8-bit ports (direction programmable)  
Single +5 Volt supply read operation  
U.V. Erasable  
40 pin DIP package

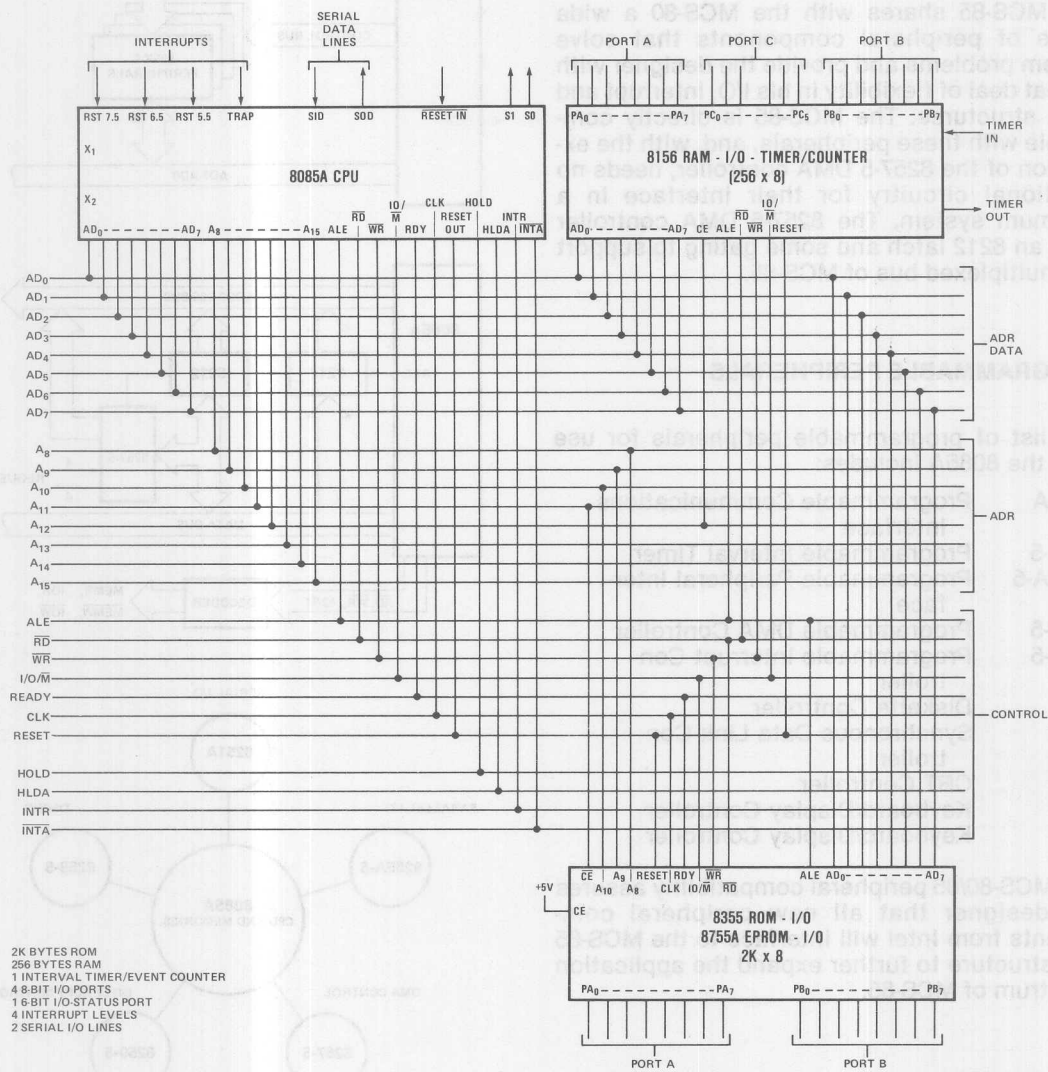
One of the most important advances made with the MCS-85 is the socket-compatibility of the 8355 and 8755A components. This allows the systems designer to develop and debug in erasable PROM and then, when satisfied, switch over to mask-programmed ROM 8355 with no performance degradation or board layout. It also allows quick prototype production for market impact without going to a compromise solution.



## SYSTEM EXPANSION

Each of these peripheral components has features that allow a small to medium system to be constructed without the addition of buffers and decoders to maintain the lowest possible component count.

# INTRODUCTION TO MCS-85™



## MCS-85™ BASIC SYSTEM

## INTERFACING TO MCS-80/85 PROGRAMMABLE PERIPHERAL COMPONENTS

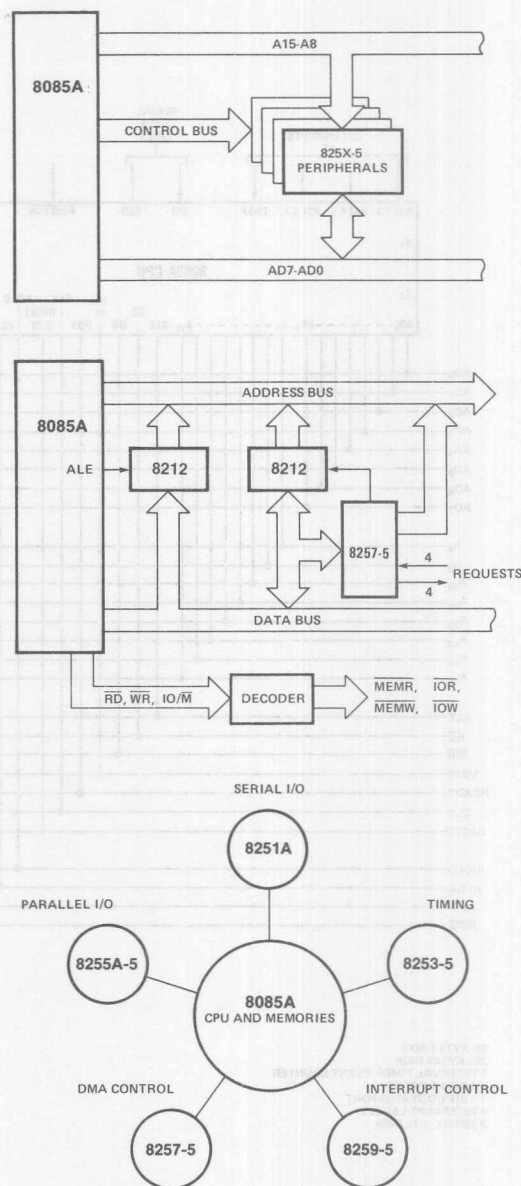
The MCS-85 shares with the MCS-80 a wide range of peripheral components that solve system problems and provide the designer with a great deal of flexibility in his I/O, Interrupt and DMA structures. The MCS-85 is directly compatible with these peripherals, and, with the exception of the 8257-5 DMA controller, needs no additional circuitry for their interface in a minimum system. The 8257-5 DMA controller uses an 8212 latch and some gating to support the multiplexed bus of MCS-85.

## PROGRAMMABLE PERIPHERALS

The list of programmable peripherals for use with the 8085A includes:

8251A	Programmable Communications Interface
8253-5	Programmable Interval Timer
8255A-5	Programmable Peripheral Interface
8257-5	Programmable DMA Controller
8259-5	Programmable Interrupt Controller
8271	Diskette Controller
8273	Synchronous Data Link Controller
8275	CRT Controller
8278	Keyboard/Display Controller
8279	Keyboard/Display Controller

The MCS-80/85 peripheral compatibility assures the designer that all new peripheral components from Intel will interface to the MCS-85 bus structure to further expand the application spectrum of MCS-85.



## INTERFACING TO STANDARD MEMORY

The MCS-85 was designed to support the full range of system configurations from small 3 chip applications to large memory and I/O applications. The 8085A CPU issues advanced READ/WRITE status signals (S0, S1, and IO/M) so that, in the case of large systems, these signals could be used to simplify bus arbitration logic and dynamic RAM refresh circuitry.

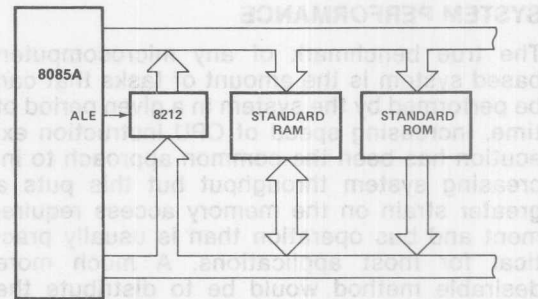
In large, memory-intensive systems, standard memory devices may provide a more cost-effective solution than do the special 8155 and 8355 devices, especially where few I/O lines are required.

## DEMULTIPLEXING THE BUS

In order to interface standard memory components such as Intel® 2114, 2142, 2716, 2316E, 2104A and 2117 the MCS-85 bus must be "demultiplexed". This is accomplished by connecting an Intel® 8212 latch to the data bus and strobing the latch with the ALE signal from the 8085A CPU. The ALE signal is issued to indicate that the multiplexed bus contains the lower 8-bits of the address. The 8212 latches this information so that a full 16-bit address is available to interface standard memory components.

## USE OF 8212

Large, memory intensive systems are usually multi-card implementations and require some form of TTL buffering to provide necessary current and voltage levels. Frequently, 8212s are used for this purpose. The 8212 has the advantage of being able to latch and demultiplex the address bus and provide extra address drive capability at the same time.



## SYSTEM PERFORMANCE

The true benchmark of any microcomputer-based system is the amount of tasks that can be performed by the system in a given period of time. Increasing speed of CPU instruction execution has been the common approach to increasing system throughput but this puts a greater strain on the memory access requirement and bus operation than is usually practical for most applications. A much more desirable method would be to distribute the task-load to peripheral devices.

## DISTRIBUTED PROCESSING

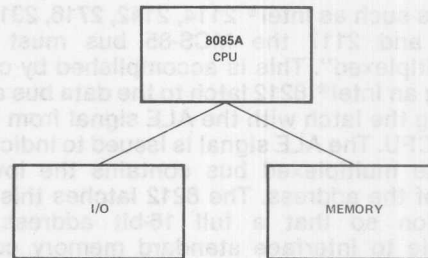
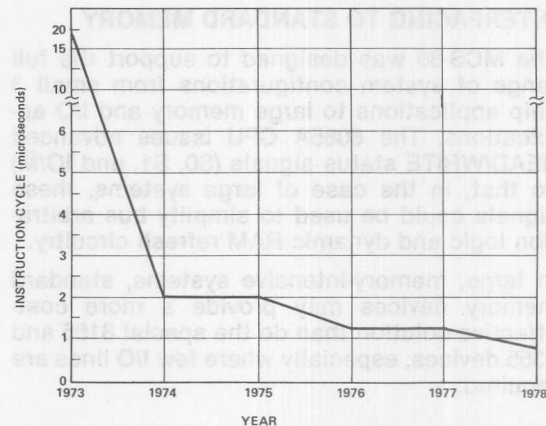
The concept of distributed task processing is not new to the computer designer, but until recently little if any task distribution was available to the microcomputer user. The use of the new programmable MCS-80/85 peripherals can relieve the central processor of many of the bookkeeping I/O and timing tasks that would otherwise have to be handled by system software.

## INSTRUCTION CYCLE/ACCESS TIME

The basic instruction cycle of the 8085A is 1.3 microseconds, the same speed as the 8080A-1. A close look at the MCS-85 bus operation shows that the access requirement for this speed is only 575 nanoseconds. The MCS-80™ access requirements for this speed would be under 300 nanoseconds. This illustrates the efficiency and improved timing margins of the MCS-85 bus structure. The new 8085A-2, a high-speed selected version of the 8085A with a .8 microsecond instruction cycle, provides a 60% performance improvement over the standard 8085A.

## CONCLUSIONS: THROUGHPUT/COST

When a total system throughput/cost analysis is taken, the MCS-85 system with its advanced processor will yield the most cost-effective, reliable and producible system.



## CHAPTER 2

### Functional Description

**MCS  
85**

**MCS  
85**

**MCS  
85**



## CHAPTER 2 FUNCTIONAL DESCRIPTION

### 2.1 WHAT THE 8085A IS

The 8085A is an 8-bit general-purpose microprocessor that is very cost-effective in small systems because of its extraordinarily low hardware overhead requirements. At the same time it is capable of accessing up to 64K bytes of memory and has status lines for controlling large systems.

### 2.2 WHAT'S IN THE 8085A

In the 8085A microprocessor are contained the functions of clock generation, system bus control, and interrupt priority selection, in addition to execution of the instruction set. (See Figure 2-1.) The 8085A transfers data on an 8-bit, bi-directional 3-state bus ( $AD_{0-7}$ ) which is time-multiplexed so as to also transmit the eight lower-order address bits. An additional eight lines ( $A_{8-15}$ ) expand the MCS-85 system memory addressing capability to 16 bits, thereby allowing 64K bytes of memory to be accessed directly by the CPU. The 8085A CPU (central processing unit) generates control signals that can be used to select appropriate external devices and

functions to perform READ and WRITE operations and also to select memory or I/O ports. The 8085A can address up to 256 different I/O locations. These addresses have the same numerical values (00 through FFH) as the first 256 memory addresses; they are distinguished by means of the  $IO/\overline{M}$  output from the CPU. You may also choose to address I/O ports as memory locations (i.e., memory-map the I/O, Section 3.2).

#### 2.2.1 Registers

The 8085A, like the 8080, is provided with internal 8-bit registers and 16-bit registers. The 8085A has eight addressable 8-bit registers. Six of them can be used either as 8-bit registers or as 16-bit register pairs. Register pairs are treated as though they were single, 16-bit registers; the high-order byte of a pair is located in the first register and the low-order byte is located in the second. In addition to the register pairs, the 8085A contains two more 16-bit registers.

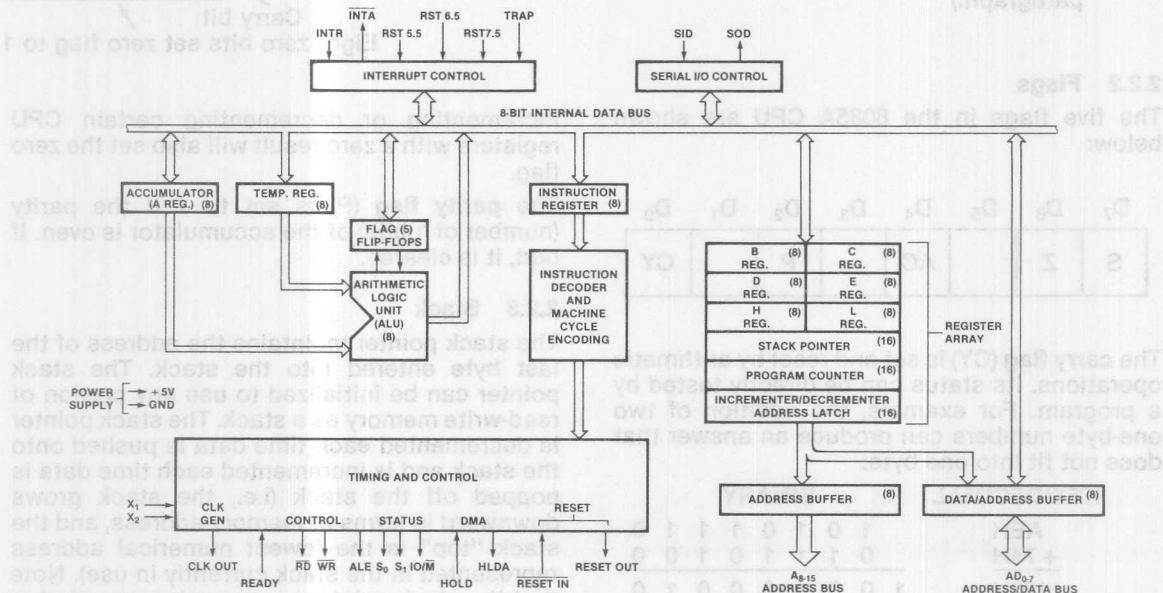


FIGURE 2-1 8085A CPU FUNCTIONAL BLOCK DIAGRAM

## FUNCTIONAL DESCRIPTION

The 8085A's CPU registers are distinguished as follows:

- The **accumulator** (ACC or A Register) is the focus of all of the accumulator instructions (Table 4-1), which include arithmetic, logic, load and store, and I/O instructions. It is an 8-bit register only. (However, see **Flags**, in this list.)
- The **program counter** (PC) always points to the memory location of the next instruction to be executed. It always contains a 16-bit address.
- **General-purpose registers** BC, DE, and HL may be used as six 8-bit registers or as three 16-bit registers, interchangeably, depending on the instruction being performed. HL functions as a **data pointer** to reference memory addresses that are either the sources or the destinations in a number of instructions. A smaller number of instructions can use BC or DE for indirect addressing.
- The **stack pointer** (SP) is a special data pointer that always points to the stack top (next available stack address). It is an indivisible 16-bit register.
- The **flag register** contains five one-bit flags, each of which records processor status information and may also control processor operation. (See following paragraph.)

### 2.2.2 Flags

The five flags in the 8085A CPU are shown below:

D <sub>7</sub>	D <sub>6</sub>	D <sub>5</sub>	D <sub>4</sub>	D <sub>3</sub>	D <sub>2</sub>	D <sub>1</sub>	D <sub>0</sub>
S	Z		AC		P		CY

The **carry flag** (CY) is set and reset by arithmetic operations. Its status can be directly tested by a program. For example, the addition of two one-byte numbers can produce an answer that does not fit into one byte:

HEXADECIMAL	BINARY
AEH	1 0 1 0 1 1 1 0
+ 74H	0 1 1 1 0 1 0 0
122H	1 0 0 1 0 0 0 1 0

↑  
Carry bit sets carry flag to 1

An addition operation that results in an overflow out of the high-order bit of the accumulator sets the carry flag. An addition operation that does not result in an overflow clears the carry flag. (See 8080/8085 Assembly Language Programming Manual for further details.) The carry flag also acts as a "borrow" flag for subtract operations.

The **auxiliary carry flag** (AC) indicates overflow out of bit 3 of the accumulator in the same way that the carry flag indicates overflow out of bit 7. This flag is commonly used in BCD (binary coded decimal) arithmetic.

The **sign flag** is set to the condition of the most significant bit of the accumulator following the execution of arithmetic or logic instructions. These instructions use bit 7 of data to represent the sign of the number contained in the accumulator. This permits the manipulation of numbers in the range from -128 to +127.

The **zero flag** is set if the result generated by certain instructions is zero. The zero flag is cleared if the result is not zero. A result that has a carry but has a zero answer byte in the accumulator will set both the carry flag and the zero flag. For example,

HEXADECIMAL	BINARY
A7H	1 0 1 0 0 1 1 1
+ 59H	+ 0 1 0 1 1 0 0 1
100H	1 0 0 0 0 0 0 0 0

↑  
Carry bit  
Eight zero bits set zero flag to 1

Incrementing or decrementing certain CPU registers with a zero result will also set the zero flag.

The **parity flag** (P) is set to 1 if the parity (number of 1-bits) of the accumulator is even. If odd, it is cleared.

### 2.2.3 Stack

The stack pointer maintains the address of the last byte entered into the stack. The stack pointer can be initialized to use any portion of read-write memory as a stack. The stack pointer is decremented each time data is pushed onto the stack and is incremented each time data is popped off the stack (i.e., the stack grows downward in terms of memory address, and the stack "top" is the lowest numerical address represented in the stack currently in use). Note that the stack pointer is always incremented or decremented by two bytes since all stack operations apply to register pairs.

## 2.2.4 Arithmetic-Logic Unit (ALU)

The ALU contains the accumulator and the flag register (described in Sections 2.2.1 and 2.2.2) and some temporary registers that are inaccessible to the programmer.

Arithmetic, logic, and rotate operations are performed by the ALU. The results of these operations can be deposited in the accumulator, or they can be transferred to the internal data bus for use elsewhere.

## 2.2.5 Instruction Register and Decoder

During an instruction fetch, the first byte of an instruction (containing the opcode) is transferred from the internal bus to the 8-bit instruction register. (See Figure 2-1.) The contents of the instruction register are, in turn, available to the instruction decoder. The output of the decoder, gated by timing signals, controls the registers, ALU, and data and address buffers. The outputs of the instruction decoder and internal clock generator generate the state and machine cycle timing signals.

## 2.2.6 Internal Clock Generator

The 8085A CPU incorporates a complete clock generator on its chip, so it requires only the addition of a quartz crystal to establish timing for its operation. (It will accept an external clock input at its  $X_1$  input instead, however.) A suitable crystal for the standard 8085A must be parallel-resonant at a fundamental of 6.25 MHz or less, twice the desired internal clock frequency. The 8085A-2 will operate with crystal of up to 10 MHz. The functions of the 8085A internal clock generator are shown in Figure 2-2. A Schmitt trigger is used interchangeably as oscillator or

as input conditioner, depending upon whether a crystal or an external source is used. The clock circuitry generates two nonoverlapping internal clock signals,  $\phi_1$  and  $\phi_2$  (see Figure 2-2).  $\phi_1$  and  $\phi_2$  control the internal timing of the 8085A and are not directly available on the outside of the chip. The external pin CLK is a buffered, inverted version of  $\phi_1$ . CLK is half the frequency of the crystal input signal and may be used for clocking other devices in the system.

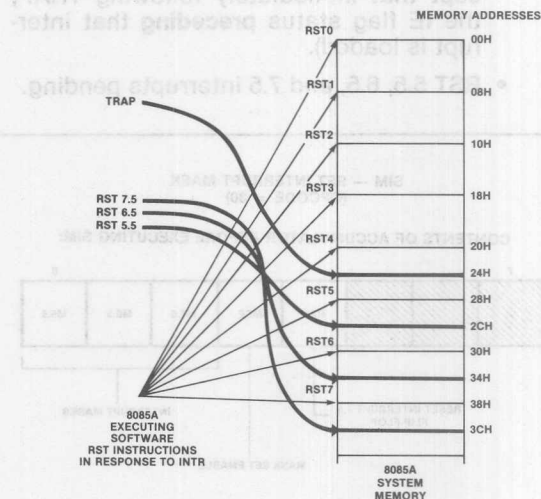
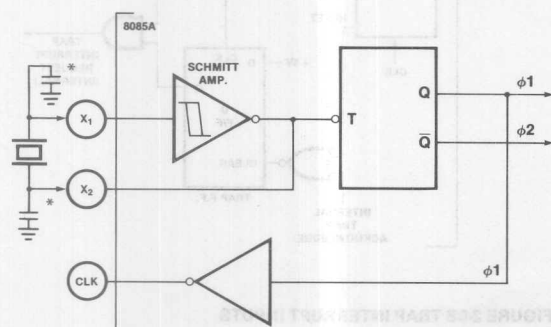


FIGURE 2-3 8085A HARDWARE AND SOFTWARE RST BRANCH LOCATIONS



\*EXTERNAL CAPACITORS REQUIRED ONLY FOR CRYSTAL FREQUENCIES  $\leq 4$ MHz.

FIGURE 2-2 8085A CLOCK LOGIC

## 2.2.7 Interrupts

The five hardware interrupt inputs provided in the 8085A are of three types. INTR is identical with the 8080A INT line in function; i.e., it is maskable (can be enabled or disabled by EI or DI software instructions), and causes the CPU to fetch in an RST instruction, externally placed on the data bus, which vectors a branch to any one of eight fixed memory locations (Restart addresses). (See Figure 2-3.) INTR can also be controlled by the 8259 programmable interrupt controller, which generates CALL instructions instead of RSTs, and can thus vector operation of the CPU to a preprogrammed subroutine located anywhere in your system's memory map. The RST 5.5, RST 6.5, and RST 7.5 hardware interrupts are different in function in that they are maskable through the use of the SIM

## FUNCTIONAL DESCRIPTION

instruction, which enables or disables these interrupts by clearing or setting corresponding mask flags based on data in the accumulator. (See Figure 2-4.) You may read the status of the interrupt mask previously set by performing a RIM instruction. Its execution loads into the accumulator the following information. (See Figure 2-5.)

- Current interrupt mask status for the RST 5.5, 6.5, and 7.5 hardware status.
- Current interrupt enable flag status (except that immediately following TRAP, the IE flag status **preceding** that interrupt is loaded).
- RST 5.5, 6.5, and 7.5 interrupts pending.

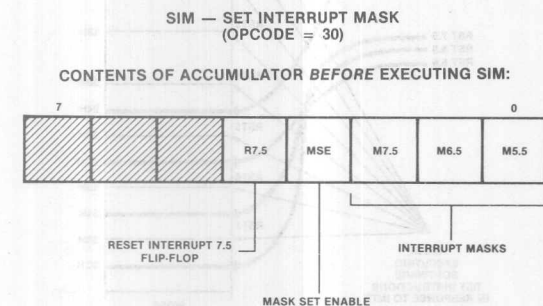


FIGURE 2-4 INTERRUPT MASKS SET USING SIM INSTRUCTION

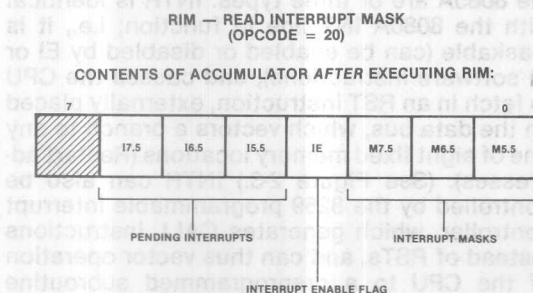


FIGURE 2-5 RIM — READ INTERRUPT MASK

RST 5.5, 6.5, and 7.5 are also subject to being enabled or disabled by the EI and DI instructions, respectively. INTR, RST 5.5, and RST 6.5 are level-sensitive, meaning that these inputs may be acknowledged by the processor when they are held at a high level. RST 7.5 is edge-sensitive, meaning that an internal flip-flop in the 8085A registers the occurrence of an interrupt the instant a rising edge appears on the RST 7.5 input line. This input need not be held high; the flip-flop will remain set until it is cleared by one of three possible actions:

- The 8085A responds to the interrupt, and sends an internal reset signal to the RST 7.5 flip-flop. (See Figure 2-6A.)

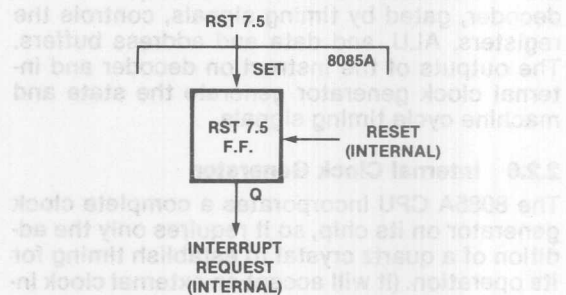


FIGURE 2-6A RST 7.5 FLIP FLOP

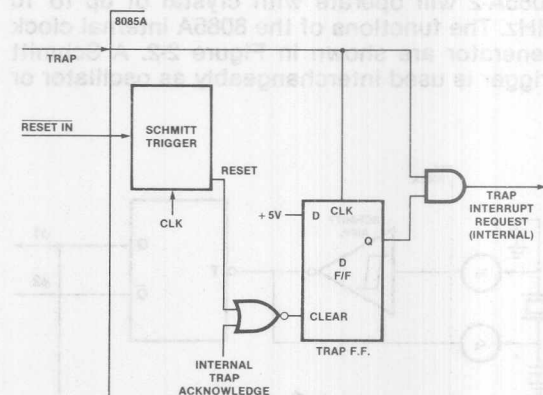


FIGURE 2-6B TRAP INTERRUPT INPUTS

FIGURE 2-6 RST 7.5 AND TRAP INTERRUPT INPUTS

## FUNCTIONAL DESCRIPTION

- The 8085A, before responding to the RST 7.5 interrupt, receives a **RESET IN** signal from an external source; this also activates the internal reset.
- The 8085A executes a SIM instruction, with accumulator bit 4 previously set to 1. (See Figure 2-4.)

The third type of hardware interrupt is TRAP. This input is not subject to any mask or interrupt enable/disable instruction. The receipt of a positive-going edge on the TRAP input triggers the processor's hardware interrupt sequence, but the pulse must be held high until acknowledged internally (see Figure 2-6B).

The sampling of all interrupts occurs on the descending edge of CLK, one cycle before the end of the instruction in which the interrupt input is activated. To be recognized, a valid interrupt must occur at least 160 ns before sampling time in the 8085A, or 150 ns in the 8085A-2. This means that to guarantee being recognized, RST 5.5 and 6.5 and TRAP need to be held on for at least 17 clock states plus 160 ns (150 for 8085A-2), assuming that the interrupt might arrive just barely too late to be acknowledged during a particular instruction, and that the following instruction might be an 18-state CALL. This timing assumes no WAIT or HOLD cycles are used.

The way interrupt masks are set and read is described in Chapter 4 under the RIM (read in-

terrupt mask) and SIM (set interrupt mask) instruction listings. Interrupt functions and their priorities are shown in the table that follows.

Name	Priority	Address (1) Branched to when inter- rupt occurs	Type Trigger
TRAP	1	24H	Rising edge AND high level until sampled
RST 7.5	2	3CH	Rising edge (latched)
RST 6.5	3	34H	High level until sam- pled
RST 5.5	4	2CH	High level until sam- pled
INTR	5	(2)	High level until sam- pled

### NOTES:

- (1) In the case of TRAP and RST 5.5-7.5, the contents of the Program Counter are pushed onto the stack before the branch occurs.
- (2) Depends on the instruction that is provided to the 8085A by the 8259 or other circuitry when the interrupt is acknowledged.

### 2.2.8 Serial Input and Output

The SID and SOD pins help to minimize chip count in small systems by providing for easy interface to a serial port using software for timing and for coding and decoding of the data. Each time a RIM instruction is executed, the status of the SID pin is read into bit 7 of the accumulator. RIM is thus a dual-purpose instruction. (See Chapter 4.) In similar fashion, SIM is used to latch bit 7 of the accumulator out to the SOD output via an internal flip-flop, providing that bit 6 of the accumulator is set to 1. (See Figure 2-7.) Section 2.3.8 describes SID and SOD timing.

SID can also be used as a general purpose TEST input and SOD can serve as a one-bit control output.

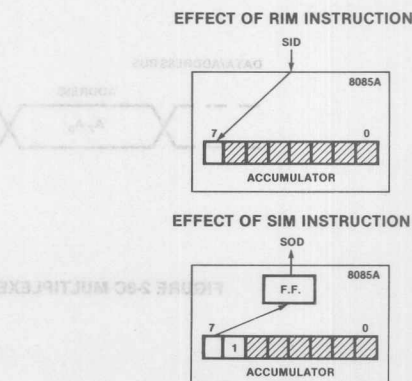


FIGURE 2-7 EFFECT OF RIM AND SIM INSTRUCTIONS ON SERIAL DATA LINES

The 8085A CPU generates signals that tell peripheral devices what type of information is on the multiplexed Address/Data bus and from that point on the operation is almost identical to the MCS-80™ CPU Group. A multiplexed bus structure was chosen because it freed device pins so that more functions could be integrated on the 8085A and other components of the family. The multiplexed bus is designed to allow complete compatibility to existing peripheral

access requirements. (See Figure 2-8.)

To enhance the system integration of MCS-85, several special components with combined memory and I/O were designed. These new devices directly interface to the multiplexed bus of the 8085A. The pin locations of the 8085A and the special peripheral components are assigned to minimize PC board area and to allow for efficient layout. The details on peripheral components are contained in subsequent paragraphs of this chapter and in Chapters 5 and 6.

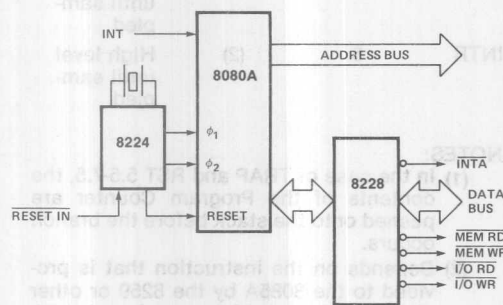


FIGURE 2-8A MCS-80™ CPU GROUP

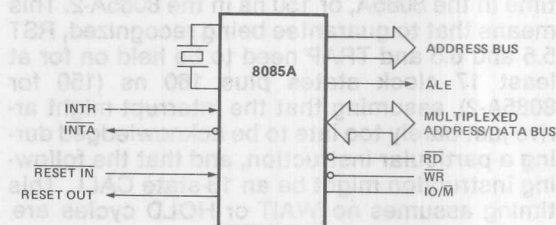


FIGURE 2-8B MCS-85™ CPU/8085A (MCS-80 COMPATIBLE FUNCTIONS)

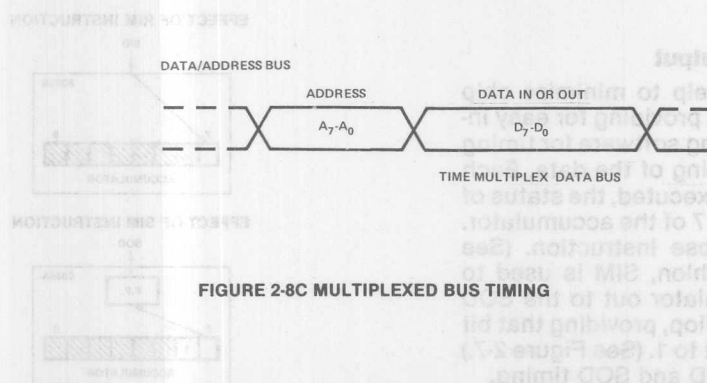


FIGURE 2-8C MULTIPLEXED BUS TIMING

FIGURE 2-8 BASIC CPU FUNCTIONS

## 2.3.1 Multiplexed Bus Cycle Timing

The execution of any 8085A program consists of a sequence of READ and WRITE operations, of which each transfers a byte of data between the 8085A and a particular memory or I/O address. These READ and WRITE operations are the only communication between the processor and the other components, and are all that is necessary to execute any instruction or program.

Each READ or WRITE operation of the 8085A is referred to as a machine cycle. The execution of each instruction by the 8085A consists of a sequence of from one to five machine cycles, and each machine cycle consists of a minimum of from three to six clock cycles (also referred to as T states). Consider the case of the Store Accumulator Direct (STA) instruction, shown in Figure 2-9. The STA instruction causes the contents of the accumulator to be stored at the direct address specified in the second and third bytes of the instruction. During the first machine cycle ( $M_1$ ), the CPU puts the contents of the program counter (PC) on the address bus and performs a MEMORY READ cycle to read from memory the opcode of the next instruction (STA). The  $M_1$  machine cycle is also referred to as the OPCODE FETCH cycle, since it fetches the operation code of the next instruction. In the fourth clock cycle ( $T_4$ ) of  $M_1$ , the CPU interprets the data read in and recognizes it as the opcode of the STA instruction. At this point the

CPU knows that it must do three more machine cycles (two MEMORY READs and one MEMORY WRITE) to complete the instruction.

The 8085A then increments the program counter so that it points to the next byte of the instruction and performs a MEMORY READ machine cycle ( $M_2$ ) at address (PC + 1). The accessed memory places the addressed data on the data bus for the CPU. The 8085A temporarily stores this data (which is the low-order byte of the direct address) internally in the CPU. The 8085A again increments the program counter to location (PC + 2) and reads from memory ( $M_3$ ) the next byte of data, which is the high-order byte of the direct address.

At this point, the 8085A has accessed all three bytes of the STA instruction, which it must now execute. The execution consists of placing the data accessed in  $M_2$  and  $M_3$  on the address bus, then placing the contents of the accumulator on the data bus, and then performing a MEMORY WRITE machine cycle ( $M_4$ ). When  $M_4$  is finished, the CPU will fetch ( $M_1$ ) the first byte of the next instruction and continue from there.

## State Transition Sequence

As the preceding example shows, the execution of an instruction consists of a series of machine cycles whose nature and sequence is determined by the opcode accessed in the  $M_1$

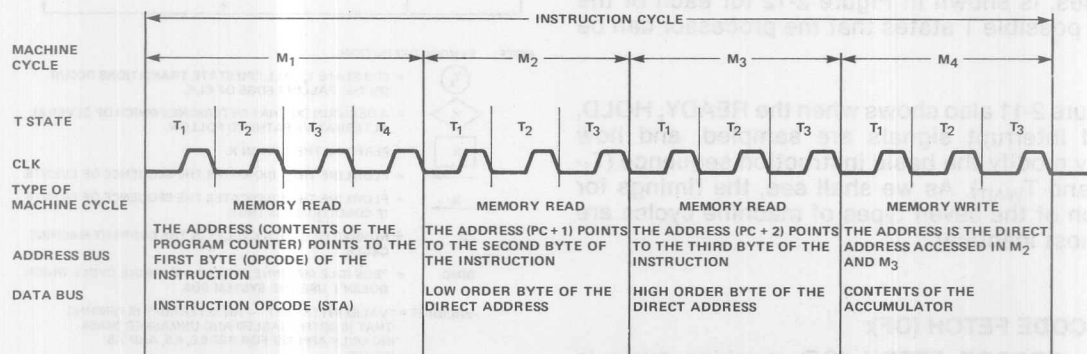


FIGURE 2-9 CPU TIMING FOR STORE ACCUMULATOR DIRECT (STA) INSTRUCTION

## FUNCTIONAL DESCRIPTION

MACHINE CYCLE		STATUS			CONTROL		
		IO/M	S <sub>1</sub>	S <sub>0</sub>	RD	WR	INTA
OPCODE FETCH (OF)		0	1	1	0	1	1
MEMORY READ (MR)		0	1	0	0	1	1
MEMORY WRITE (MW)		0	0	1	1	0	1
I/O READ (IOR)		1	1	0	0	1	1
I/O WRITE (IOW)		1	0	1	1	0	1
INTR ACKNOWLEDGE (INA)		1	1	1	1	1	0
BUS IDLE (BI): DAD		0	1	0	1	1	1
INA(RST/TRAP)		1	1	1	1	1	1
HALT		TS	0	0	TS	TS	1

0 = Logic "0" 1 = Logic "1" TS = High Impedance X = Unspecified

FIGURE 2-10 8085A MACHINE CYCLE CHART

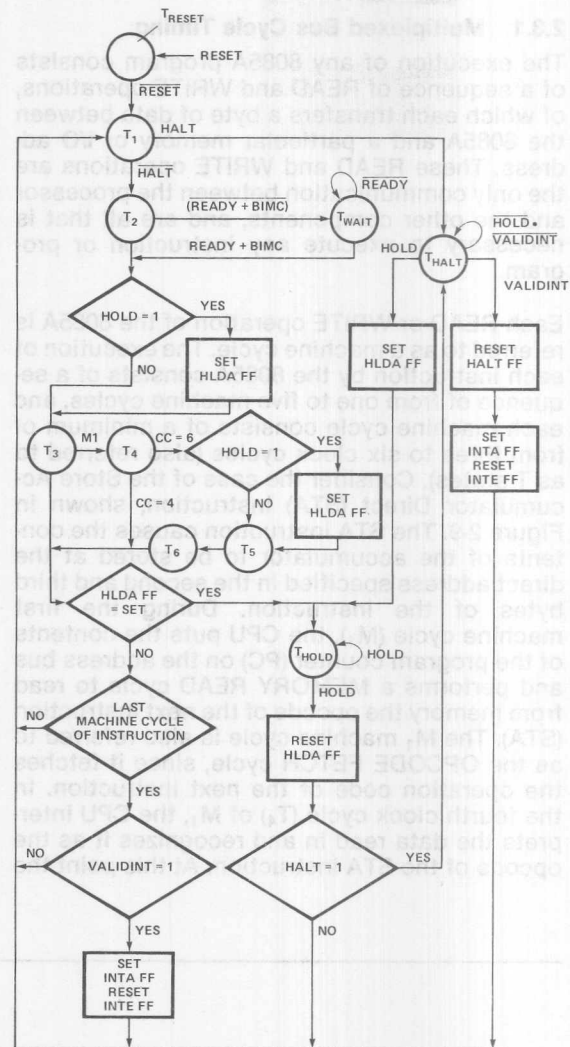
machine cycle. While no one instruction cycle will consist of more than five machine cycles, every machine cycle will be one of the seven types listed in Figure 2-10. These seven types of machine cycles can be differentiated by the state of the three status lines (IO/M, S<sub>0</sub>, and S<sub>1</sub>) and the three control signals (RD, WR, and INTA).

Most machine cycles consist of three T states, (cycles of the CLK output) with the exception of OPCODE FETCH, which normally has either four or six T states. The actual number of states required to perform any instruction depends on the instruction being executed, the particular machine cycle within the instruction cycle, and the number of WAIT and HOLD states inserted into each machine cycle through the use of the READY and HOLD inputs of the 8085A. The state transition diagram in Figure 2-11 illustrates how the 8085A proceeds in the course of a machine cycle. The state of various status and control signals, as well as the system buses, is shown in Figure 2-12 for each of the ten possible T states that the processor can be in.

Figure 2-11 also shows when the READY, HOLD, and interrupt signals are sampled, and how they modify the basic instruction sequence (T<sub>1</sub>, T<sub>6</sub> and T<sub>WAIT</sub>). As we shall see, the timings for each of the seven types of machine cycles are almost identical.

### OPCODE FETCH (OF):

The OPCODE FETCH (OF) machine cycle is unique in that it has more than three clock cycles. This is because the CPU must interpret the opcode accessed in T<sub>1</sub>, T<sub>2</sub>, and T<sub>3</sub> before it can decide what to do next.



NOTE: SYMBOL DEFINITION

- = CPU STATE T<sub>n</sub>. ALL CPU STATE TRANSITIONS OCCUR ON THE FALLING EDGE OF CLK.
- = A DECISION (X) THAT DETERMINES WHICH OF SEVERAL ALTERNATIVE PATHS TO FOLLOW.
- = PERFORM THE ACTION X.
- = FLOWLINE THAT INDICATES THE SEQUENCE OF EVENTS.
- = FLOWLINE THAT INDICATES THE SEQUENCE OF EVENTS IF CONDITION X IS TRUE.
- CC = NUMBER OF CLOCK CYCLES IN THE CURRENT MACHINE CYCLE.
- BIMC = "BUS IDLE MACHINE CYCLE" = MACHINE CYCLE WHICH DOESN'T USE THE SYSTEM BUS.
- VALIDINT = "VALID INTERRUPT" - AN INTERRUPT IS PENDING THAT IS BOTH ENABLED AND UNMASKED (MASKING ONLY APPLIES FOR RST 5.5, 6.5, AND 7.5 INPUTS).
- HLDA FF = INTERNAL HOLD ACKNOWLEDGE FLIP FLOP. NOTE THAT THE 8085A SYSTEM BUSES ARE 3-STATE ONE CLOCK CYCLE AFTER THE HLDA FLIP FLOP IS SET.

FIGURE 2-11 8085A CPU STATE TRANSITION

## FUNCTIONAL DESCRIPTION

Machine State	Status & Buses				Control		
	S1,S0	IO/ $\overline{M}$	A <sub>8</sub> -A <sub>15</sub>	AD <sub>0</sub> -AD <sub>7</sub>	$\overline{RD}$ ,WR	$\overline{INTA}$	ALE
T <sub>1</sub>	X	X	X	X	1	1	1 <sup>†</sup>
T <sub>2</sub>	X	X	X	X	X	X	0
T <sub>WAIT</sub>	X	X	X	X	X	X	0
T <sub>3</sub>	X	X	X	X	X	X	0
T <sub>4</sub>	1	0*	X	TS	1	1	0
T <sub>5</sub>	1	0*	X	TS	1	1	0
T <sub>6</sub>	1	0*	X	TS	1	1	0
T <sub>RESET</sub>	X	TS	TS	TS	TS	1	0
T <sub>HALT</sub>	0	TS	TS	TS	TS	1	0
T <sub>HOLD</sub>	X	TS	TS	TS	TS	1	0

0 = Logic "0" 1 = Logic "1" TS = High Impedance X = Unspecified

<sup>†</sup> ALE not generated during 2nd and 3rd machine cycles of DAD instruction.

\*IO/ $\overline{M}$  = 1 during T<sub>4</sub>-T<sub>6</sub> states of RST and INA cycles.

**FIGURE 2-12 8085A MACHINE STATE CHART**

Figure 2-13 shows the timing relationships for an OF machine cycle. The particular instruction illustrated is DCX, whose timing for OF differs from other instructions in that it has six T states, while some instructions require only four T states for OF. In this discussion, as well as the following discussions, only the relative timing of the signals will be discussed; for the actual timings, refer to the data sheets of the individual parts in Chapters 5 and 6.

The first thing that the 8085A does at the beginning of every machine cycle is to send out three status signals (IO/ $\overline{M}$ , S1, S0) that define what type of machine cycle is about to take place. The IO/ $\overline{M}$  signal identifies the machine cycle as being either a memory reference or input/output operation. The S1 status signal identifies whether the cycle is a READ or WRITE operation. The S0 and S1 status signals can be used together (see Figure 2-10) to identify READ, WRITE, or OPCODE FETCH machine cycles as well as the HALT state. Referring to Figure 2-13, the 8085A will send out IO/ $\overline{M}$  = 0, S1 = 1, S0 = 1 at the beginning of the machine cycle to identify it as a READ from a memory location to obtain an opcode; in other words, it identifies the machine cycle as an OPCODE FETCH cycle.

The 8085A also sends out a 16-bit address at the beginning of every machine cycle to identify the particular memory location or I/O port that the machine cycle applies to. In the case of an OF cycle, the contents of the program counter is placed on the address bus. The high order byte (PCH) is placed on the A<sub>8</sub>-A<sub>15</sub> lines, where it will stay until at least T<sub>4</sub>. The low order byte (PCL) is placed on the AD<sub>0</sub>-AD<sub>7</sub> lines, whose three-state drivers are enabled if not found already on. Unlike the upper address lines, however, the information on the lower address lines will remain there for only one clock cycle, after which the drivers will go to their high impedance state, indicated by a dashed line in Figure 2-13. This is necessary because the AD<sub>0</sub>-AD<sub>7</sub> lines are time multiplexed between the address and data buses. During T<sub>1</sub> of every machine cycle, AD<sub>0</sub>-AD<sub>7</sub> output the lower 8-bits of address after which AD<sub>0</sub>-AD<sub>7</sub> will either output the desired data for a WRITE operation or the drivers will float (as is the case for the OF cycle), allowing the external device to drive the lines for a READ operation.

Since the address information on AD<sub>0</sub>-AD<sub>7</sub> is of a transitory nature, it must be latched either internally in special multiplexed-bus components like the 8155 or externally in parts like the 8212 8-bit latch. (See Chapter 3.) The 8085A provides a special timing signal, ADDRESS LATCH ENABLE (ALE), to facilitate the latching of A<sub>0</sub>-A<sub>7</sub>; ALE is present during T<sub>1</sub> of every machine cycle.

After the status signals and address have been sent out and the AD<sub>0</sub>-AD<sub>7</sub> drivers have been disabled, the 8085A provides a low level on  $\overline{RD}$  to enable the addressed memory device. The device will then start driving the AD<sub>0</sub>-AD<sub>7</sub> lines; this is indicated by the dashed line turning into a solid line in Figure 2-13. After a period of time (which is the access time of the memory) valid data will be present on AD<sub>0</sub>-AD<sub>7</sub>. The 8085A during T<sub>3</sub> will load the memory data on AD<sub>0</sub>-AD<sub>7</sub> into its instruction register and then raise  $\overline{RD}$  to the high level, disabling the addressed memory device. At this point, the 8085A will have finished accessing the opcode of the instruction. Since this is the first machine cycle (M<sub>1</sub>) of the instruction, the CPU will automatically step to T<sub>4</sub>, as shown in Figure 2-11.

During T<sub>4</sub>, the CPU will decode the opcode in the instruction register and decide whether to enter T<sub>5</sub> on the next clock or to start a new machine cycle and enter T<sub>1</sub>. In the case of the DCX instruction shown in Figure 2-13, it will enter T<sub>5</sub> and then T<sub>6</sub> before going to T<sub>1</sub>.

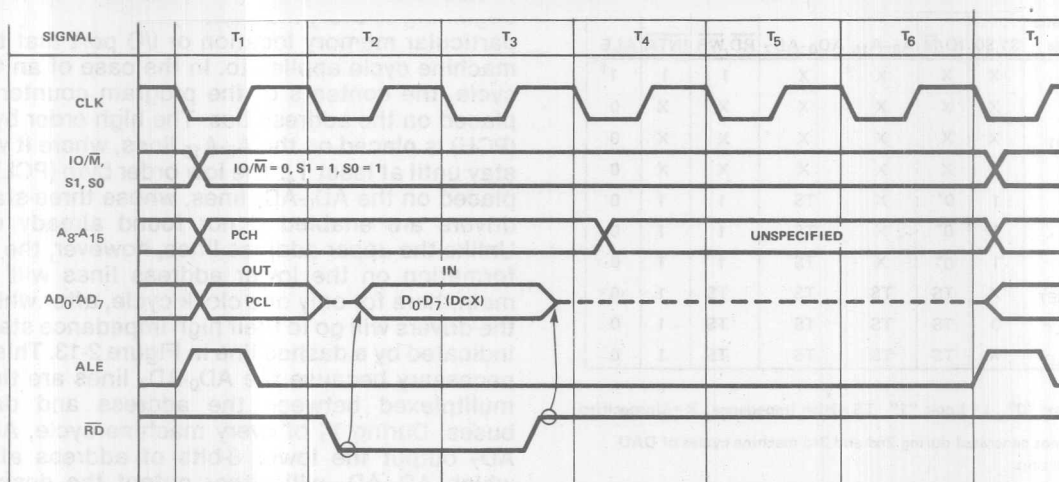


FIGURE 2-13 OPCODE FETCH MACHINE CYCLE (OF DCX INSTRUCTION)

During  $T_5$  and  $T_6$ , of DCX, the CPU will decrement the designated register. Since the  $A_8-A_{15}$  lines are driven by the address latch circuits, which are part of the incrementer/decrementer logic, the  $A_8-A_{15}$  lines may change during  $T_5$  and  $T_6$ . Because the value of  $A_8-A_{15}$  can vary during  $T_4-T_6$ , it is most important that all memory and I/O devices on the system bus qualify their selection with  $\overline{RD}$ . If they don't use  $\overline{RD}$ , they may be spuriously selected. Moreover, with a linear selection technique (Chapter 3), two or more devices could be simultaneously enabled, which could be potentially damaging. The generation of spurious addresses can also occur momentarily at address bus transitional periods in  $T_1$ . Therefore, the selection of all memory and I/O devices must be qualified with  $\overline{RD}$  or  $\overline{WR}$ . Many new memory devices like the 8155 and 8355 have the  $\overline{RD}$  input that internally is used to enable the data bus outputs, removing the need for externally qualifying the chip enable input with  $\overline{RD}$ .

Figure 2-14 is identical to Figure 2-13 with one exception, which is the use of the READY line. As we can see in Figure 2-11, when the CPU is in  $T_2$ , it examines the state of the READY line. If the READY line is high, the CPU will proceed to  $T_3$  and finish executing the instruction. If the READY line is low, however, the CPU will enter  $T_{WAIT}$  and stay there indefinitely until READY goes high. When the READY line does go high, the CPU will exit  $T_{WAIT}$  and enter  $T_3$ , in order to complete the machine cycle. As shown in

Figure 2-14, the external effect of using the READY line is to preserve the exact state of the processor signals at the end of  $T_2$  for an integral number of clock periods, before finishing the machine cycle. This "stretching" of the system timing has the further effect of increasing the allowable access time for memory or I/O devices. By inserting  $T_{WAIT}$  states, the 8085A can accommodate even the slowest of memories. Another common use of the READY line is to single-step the processor with a manual switch.

### 2.3.2 Read Cycle Timing

#### MEMORY READ (MR):

Figure 2-15 shows the timing of two successive MEMORY READ (MR) machine cycles, the first without a  $T_{WAIT}$  state and the second with one  $T_{WAIT}$  state. The timing during  $T_1-T_3$  is absolutely identical to the OPCODE FETCH machine cycle, with the exception that the status sent out during  $T_1$  is  $IO/\overline{M} = 0$ ,  $S_1 = 1$ ,  $S_0 = 0$ , identifying the cycles as a READ from a memory location. This differs from Figure 2-13 only in that  $S_0 = 1$  for an OF cycle, identifying that cycle as an OPCODE FETCH operation. Otherwise, the two cycles are identical during  $T_1-T_3$ .

A second difference occurs at the end of  $T_3$ . As shown in Figure 2-11, the CPU always goes to  $T_4$  from  $T_3$  during  $M_1$ , which is always an OF cycle. During all other machine cycles, the CPU will always go from  $T_3$  to  $T_1$  of the next machine cycle.

## FUNCTIONAL DESCRIPTION

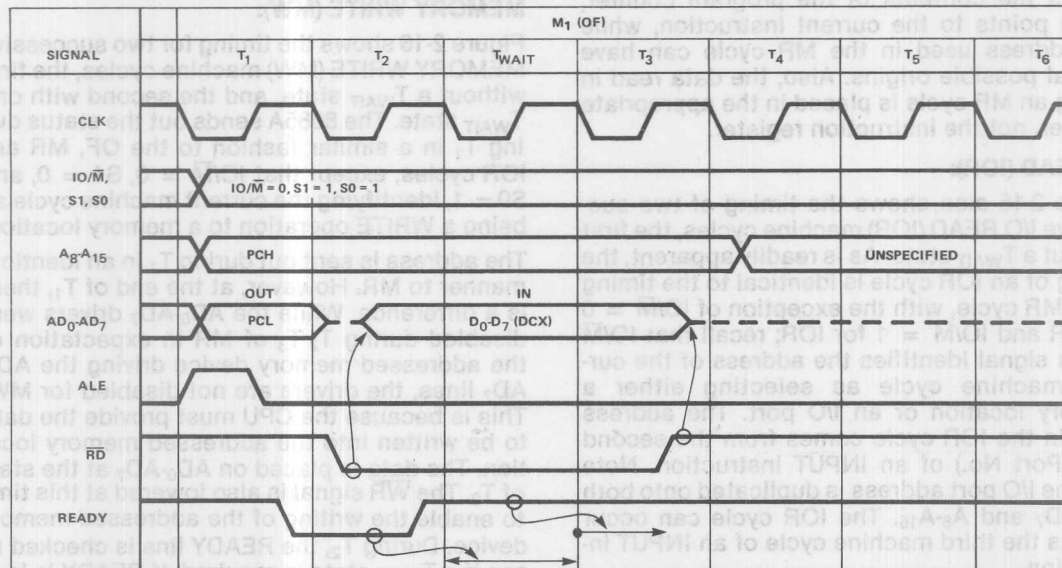


FIGURE 2-14 OPCODE FETCH MACHINE CYCLE WITH ONE WAIT STATE

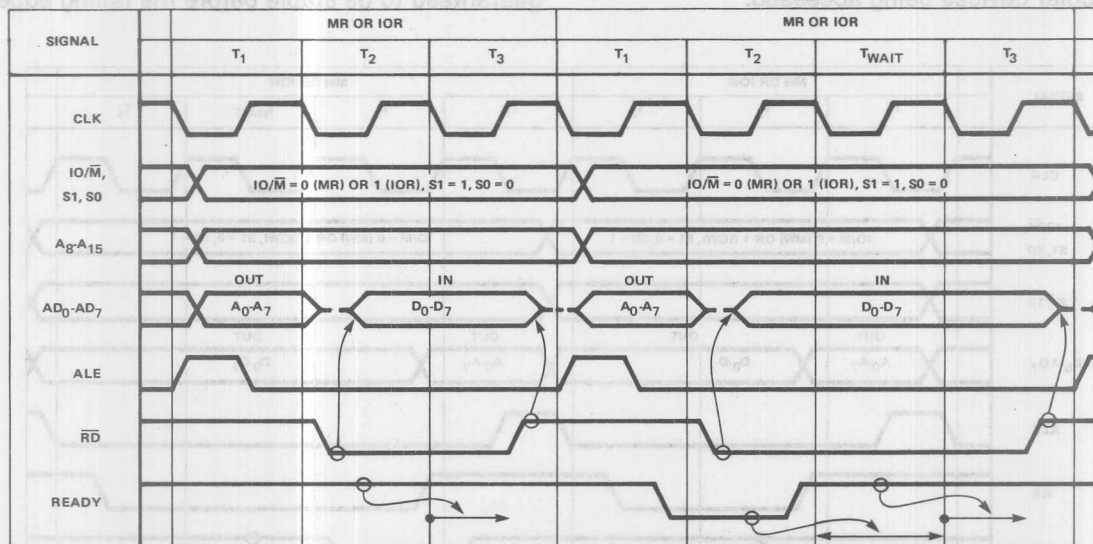


FIGURE 2-15 MEMORY READ (OR I/O READ) MACHINE CYCLES (WITH AND WITHOUT WAIT STATES)

## FUNCTIONAL DESCRIPTION

The memory address used in the OF cycle is always the contents of the program counter, which points to the current instruction, while the address used in the MR cycle can have several possible origins. Also, the data read in during an MR cycle is placed in the appropriate register, not the instruction register.

### I/O READ (IOR):

Figure 2-15 also shows the timing of two successive I/O READ (IOR) machine cycles, the first without a  $T_{WAIT}$  state. As is readily apparent, the timing of an IOR cycle is identical to the timing of an MR cycle, with the exception of  $IO/\overline{M} = 0$  for MR and  $IO/\overline{M} = 1$  for IOR; recall that  $IO/\overline{M}$  status signal identifies the address of the current machine cycle as selecting either a memory location or an I/O port. The address used in the IOR cycle comes from the second byte (Port No.) of an INPUT instruction. Note that the I/O port address is duplicated onto both  $AD_0-AD_7$  and  $A_8-A_{15}$ . The IOR cycle can occur only as the third machine cycle of an INPUT instruction.

Note that the READY signal can be used to generate  $T_{WAIT}$  states for I/O devices as well as memory devices. By gating the READY signal with the proper status lines, one could generate  $T_{WAIT}$  states for memory devices only or for I/O devices only. By gating in the address lines, one can further qualify  $T_{WAIT}$  state generation by the particular devices being accessed.

### 2.3.3 WRITE Cycle Timing

#### MEMORY WRITE (MW):

Figure 2-16 shows the timing for two successive MEMORY WRITE (MW) machine cycles, the first without a  $T_{WAIT}$  state, and the second with one  $T_{WAIT}$  state. The 8085A sends out the status during  $T_1$  in a similar fashion to the OF, MR and IOR cycles, except that  $IO/\overline{M} = 0$ ,  $S_1 = 0$ , and  $S_0 = 1$ , identifying the current machine cycle as being a WRITE operation to a memory location.

The address is sent out during  $T_1$  in an identical manner to MR. However, at the end of  $T_1$ , there is a difference. While the  $AD_0-AD_7$  drivers were disabled during  $T_2-T_3$  of MR in expectation of the addressed memory device driving the  $AD_0-AD_7$  lines, the drivers are not disabled for MW. This is because the CPU must provide the data to be written into the addressed memory location. The data is placed on  $AD_0-AD_7$  at the start of  $T_2$ . The  $\overline{WR}$  signal is also lowered at this time to enable the writing of the addressed memory device. During  $T_2$ , the READY line is checked to see if a  $T_{WAIT}$  state is required. If READY is low,  $T_{WAIT}$  states are inserted until READY goes high. During  $T_3$ , the  $\overline{WR}$  line is raised, disabling the addressed memory device and thereby terminating the WRITE operation. The contents of the address and data lines are not changed until the next  $T_1$ , which directly follows.

Note that the data on  $AD_0-AD_7$  is not guaranteed to be stable before the falling edge

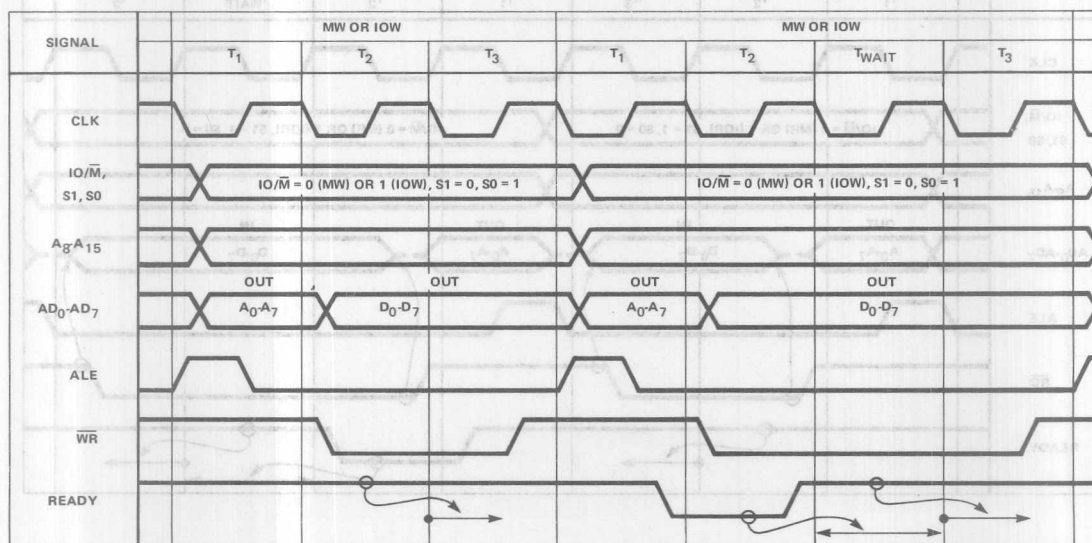


FIGURE 2-16 MEMORY WRITE (OR I/O WRITE) MACHINE CYCLES (WITH AND WITHOUT WAIT STATES)

of  $\overline{WR}$ . The  $AD_0-AD_7$  lines are guaranteed to be stable both before and after the rising edge of  $\overline{WR}$ .

## I/O WRITE (IOW):

As Figure 2-16 shows, the timing for an I/O WRITE (IOW) machine cycle is the same as an MW machine cycle except that  $IO/\overline{M} = 0$  during the MW cycle and  $IO/\overline{M} = 1$  during the IOW cycle.

As with the IOR cycle discussed previously, the address used in an IOW cycle is the I/O port number which is duplicated on both the high and low bytes of the address bus. In the case of IOW, the port number comes from the second byte of an OUTPUT instruction as the instruction is executed.

## 2.3.4 Interrupt Acknowledge (INA) Timing

Figures 2-17 and 2-18 (a continuation of 2-17) depict the course of action the CPU takes in response to a high level on the INTR line if the INTE FF (interrupt enable flip-flop) has been set

by the EI instruction. The status of the TRAP and RST pins as well as INTR is sampled during the second clock cycle before  $M_1 \cdot T_1$ . If INTR was the only valid interrupt and if INTE FF is set, then the CPU will reset INTE FF and then enter an INTERRUPT ACKNOWLEDGE (INA) machine cycle. The INA cycle is identical to an OF cycle with two exceptions.  $\overline{INTA}$  is sent out instead of  $\overline{RD}$ . Also,  $IO/\overline{M} = 1$  during INA, whereas  $IO/\overline{M} = 0$  for OF. Although the contents of the program counter are sent out on the address lines, the address lines can be ignored.

When  $\overline{INTA}$  is sent out, the external interrupt logic must provide the opcode of an instruction to execute. The opcode is placed on the data bus and read in by the processor. If the opcode is the first byte of a multiple-byte instruction, additional  $\overline{INTA}$  pulses will be provided by the 8085A to clock in the remaining bytes. RESTART and CALL instructions are the most

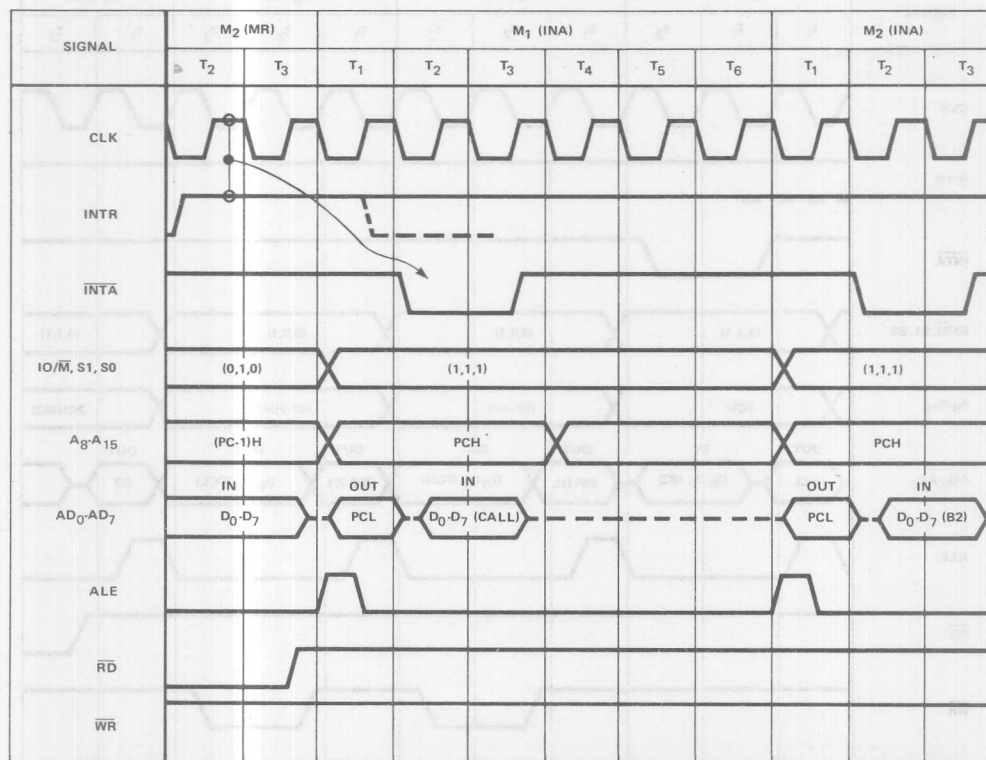


FIGURE 2-17 INTERRUPT ACKNOWLEDGE MACHINE CYCLES (WITH CALL INSTRUCTION IN RESPONSE TO INTR)

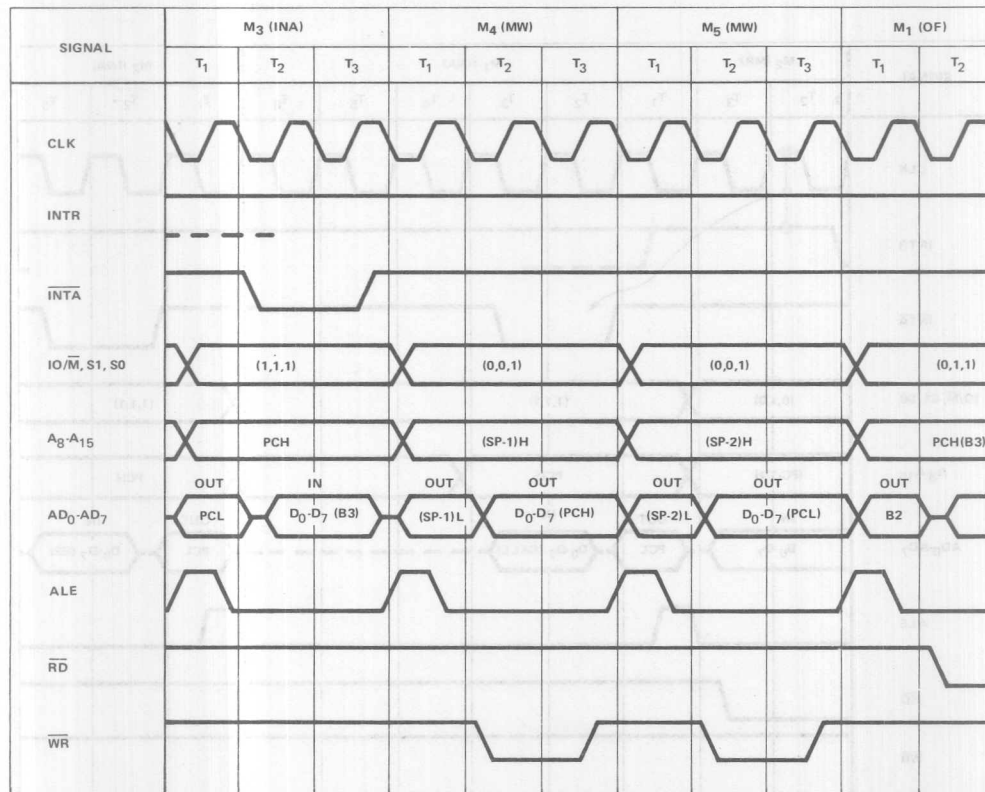
## FUNCTIONAL DESCRIPTION

logical choices, since they both force the processor to push the contents of the program counter onto the stack before jumping to a new location. In Figure 2-17 it is assumed that a CALL opcode is sent to the CPU during M<sub>1</sub>. The CALL opcode could have been placed there by a device like the 8259 programmable interrupt controller.

After receiving the opcode, the processor then decodes it and determines, in this case, that the CALL instruction requires two more bytes. The CPU therefore performs a second INA cycle (M<sub>2</sub>) to access the second byte of the instruction from the 8259. The timing of this cycle is identical to M<sub>1</sub>, except that it has only three T states. M<sub>2</sub> is followed by another INA cycle (M<sub>3</sub>) to access the third byte of the CALL instruction from the 8259.

Now that the CPU has accessed the entire instruction used to acknowledge the interrupt, it will execute that instruction. Note that any instruction could be used (except EI or DI, the instructions which enable or disable interrupts), but the RESTART and CALL instructions are the most logical choices. Also notice that the CPU inhibited the incrementing of the program counter (PC) during the three INA cycles, so that the correct PC value can be pushed onto the stack during M<sub>4</sub> and M<sub>5</sub>.

During M<sub>4</sub> and M<sub>5</sub>, the CPU performs MEMORY WRITE machine cycles to write the upper and then lower bytes of the PC onto the top of the stack. The CPU then places the two bytes accessed in M<sub>2</sub> and M<sub>3</sub> into the lower and upper bytes of the PC. This has the effect of jumping the execution of the program to the location specified by the CALL instruction.



**FIGURE 2-18 INTERRUPT ACKNOWLEDGE MACHINE CYCLES (WITH CALL INSTRUCTION IN RESPONSE TO INTR)**

## 2.3.5 Bus Idle (BI) and HALT State

Most machine cycles of the 8085A are associated with either a READ or WRITE operation. There are two exceptions to this rule. The first exception takes place during  $M_2$  and  $M_3$  of the DAD instruction. The 8085A requires six internal T states to execute a DAD instruction, but it is not desirable to have  $M_1$  be ten (four normal plus six extra) states long. Therefore, the CPU generates two extra machine cycles that do not access either the memory or the I/O. These cycles are referred to as BUS IDLE (BI) machine cycles. In the case of DAD, they are identical to MR cycles except that RD remains high and ALE is not generated. Note that READY is ignored during  $M_2$  and  $M_3$  of DAD.

The other time when the BUS IDLE machine cycle occurs is during the internal opcode generation for the RST or TRAP interrupts. Figure 2-19 illustrates the BI cycle generated in response to RST 7.5. Since this interrupt is rising-edge-triggered, it sets an internal latch; that latch is sampled at the falling edge of the next to the last T-state of the previous instruction. At this point the CPU must generate its own internal RESTART instruction which will (in subsequent machine cycles) cause the processor to push the program counter on the stack and to vector to location 3CH. To do this, it executes an OF machine cycle without issuing RD, generating the RESTART opcode instead. After  $M_1$ , the CPU continues execution normally in all respects except that the state of the READY line is ignored during the BI cycle.

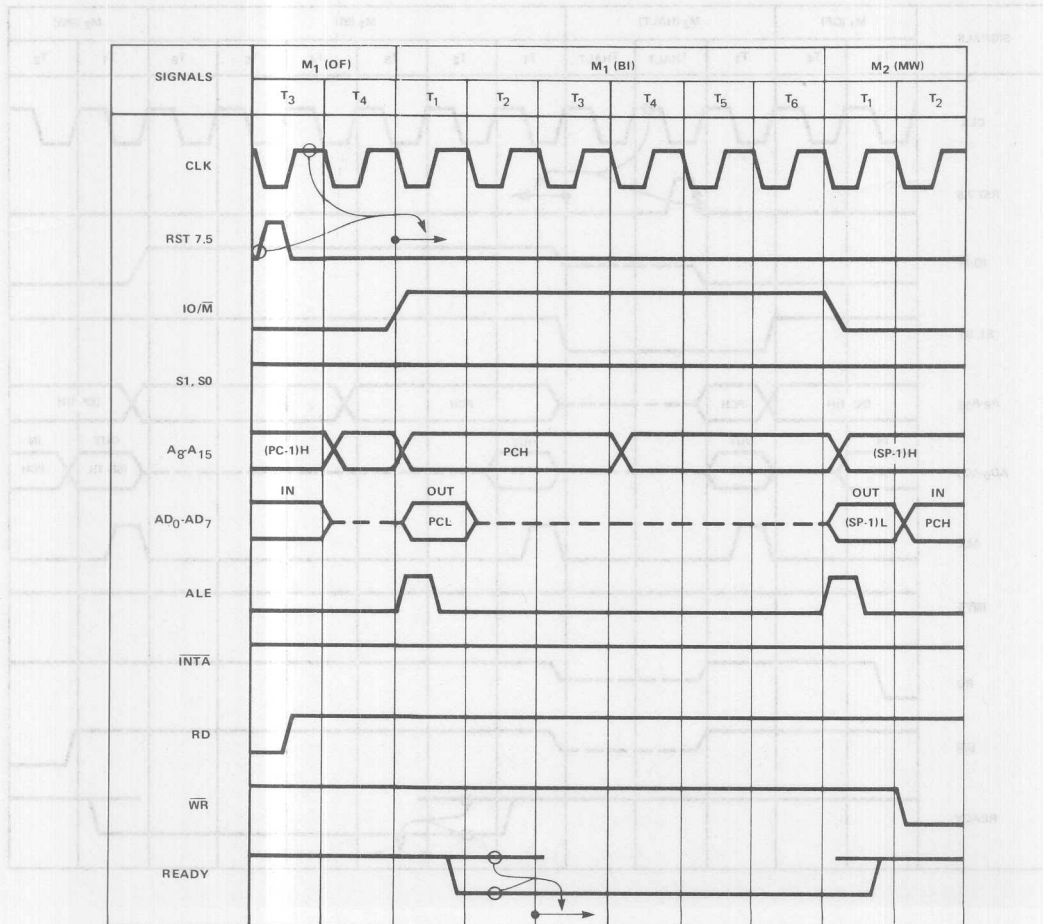


FIGURE 2-19 RST 7.5 BUS IDLE MACHINE CYCLE

## FUNCTIONAL DESCRIPTION

Figure 2-20 illustrates the BI cycle generated in response to RST 7.5 when a HALT instruction has just been executed and the CPU is in the  $T_{\text{HALT}}$  state, with its various signals floating. There are only two ways the processor can completely exit the  $T_{\text{HALT}}$  state, as shown in Figure 2-11. The first way is for RESET to occur, which always forces the 8085A to  $T_{\text{RESET}}$ . The second way to exit  $T_{\text{HALT}}$  permanently is for a valid interrupt to occur, which will cause the CPU to disable further interrupts by resetting INTE FF, and to then proceed to  $M_1 \cdot T_1$  of the next instruction. When the HOLD input is activated, the CPU will exit  $T_{\text{HALT}}$  for the duration of  $T_{\text{HOLD}}$  and then return to  $T_{\text{HALT}}$ .

In Figure 2-20 the RST 7.5 line is pulsed during  $T_{\text{HALT}}$ . Since RST 7.5 is a rising-edge-triggered interrupt, it will set an internal latch which is sampled during  $\text{CLK} = "1"$  of every  $T_{\text{HALT}}$  state (as well as during  $\text{CLK} = "1"$  two  $T$  states before any  $M_1 \cdot T_1$ .) The fact that the latched interrupt was high (assuming that INTE FF = 1 and the RST 7.5 mask = 0) will force the CPU to exit the  $T_{\text{HALT}}$  state at the end of the next CLK period, and to enter  $M_1 \cdot T_1$ .

This completes our analysis of the timing of each of the seven types of machine cycles.

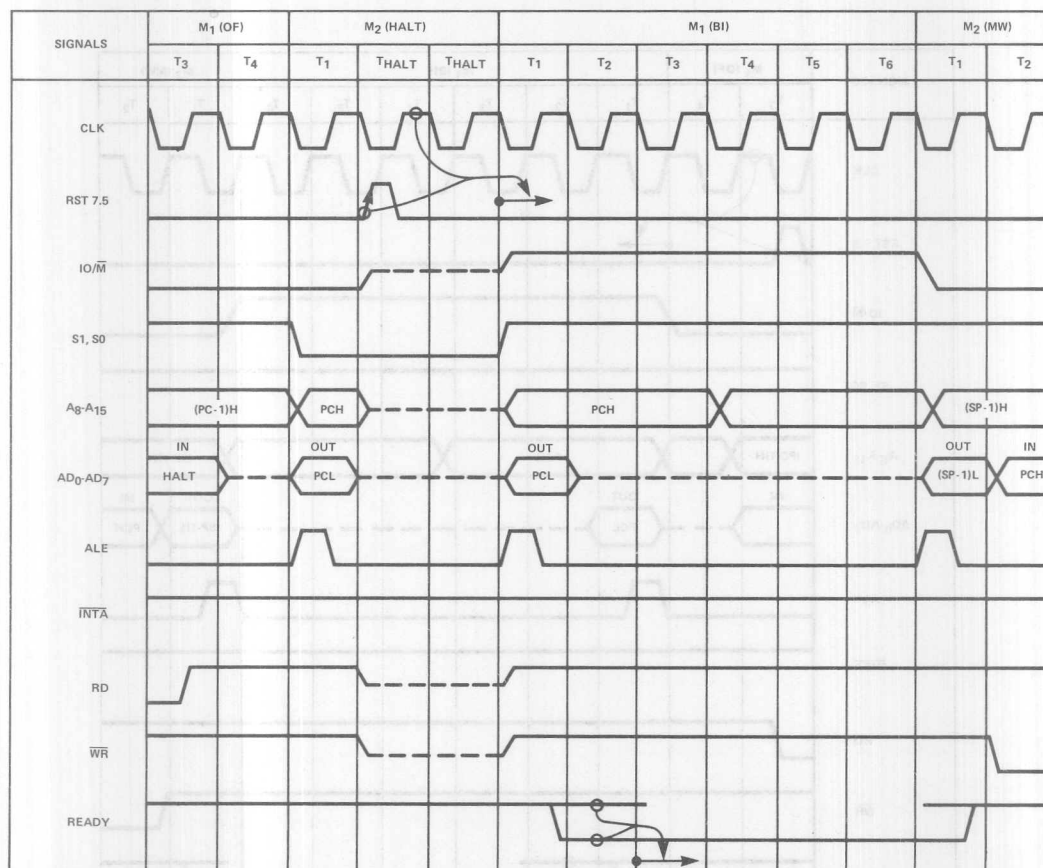


FIGURE 2-20 HALT STATE AND BUS IDLE MACHINE CYCLE  
RST 7.5 TERMINATES  $T_{\text{HALT}}$  STATE

## FUNCTIONAL DESCRIPTION

### 2.3.6 HOLD and HALT States

The 8085A uses the  $T_{HOLD}$  state to momentarily cease executing machine cycles, allowing external devices to gain control of the bus and perform DMA cycles. The processor internally latches the state of the HOLD line and the unmasked interrupts during  $CLK = "1"$  of every  $T_{HALT}$  state. If the internal latched HOLD signal is high during  $CLK = "1"$  of any  $T_{HALT}$  state, the CPU will exit  $T_{HALT}$  and enter  $T_{HOLD}$  on the following  $CLK = "1"$ . As shown in Figure 2-21 this will occur even if a valid interrupt occurs simultaneously with the HOLD signal.

The state of the HOLD and the unmasked interrupt lines is latched internally during  $CLK = 1$  of each  $T_{HOLD}$  state as well as during  $T_{HALT}$  states. If the internal latched HOLD signal is low during  $CLK = 1$ , the CPU will exit  $T_{HOLD}$  and enter  $T_{HALT}$  on the following  $CLK = 1$ .

The 8085A accepts the first unmasked, enabled interrupt sampled; thereafter, all interrupt sampling is inhibited. The interrupt thus accepted will inevitably be executed when the CPU exits the HOLD state, even at the expense of holding off higher-priority interrupts (including TRAP). (See Figure 2-22.)

When the CPU is not in  $T_{HALT}$  or  $T_{HOLD}$ , it internally latches the HOLD line only during  $CLK = 1$  of the last state before  $T_3$  ( $T_2$  or  $T_{WAIT}$ ) and during  $CLK = 1$  of the last state before  $T_5$  ( $T_4$  of a six T-state  $M_1$ ). If the internal latched HOLD signal is high during the next  $CLK = 1$ , the CPU will enter  $T_{HOLD}$  after the following clock. When the CPU is not in  $T_{HALT}$  or  $T_{HOLD}$ , it will internally latch the state of the unmasked interrupts only during  $CLK$  of the next to the last state before each  $M_1 \cdot T_1$ .

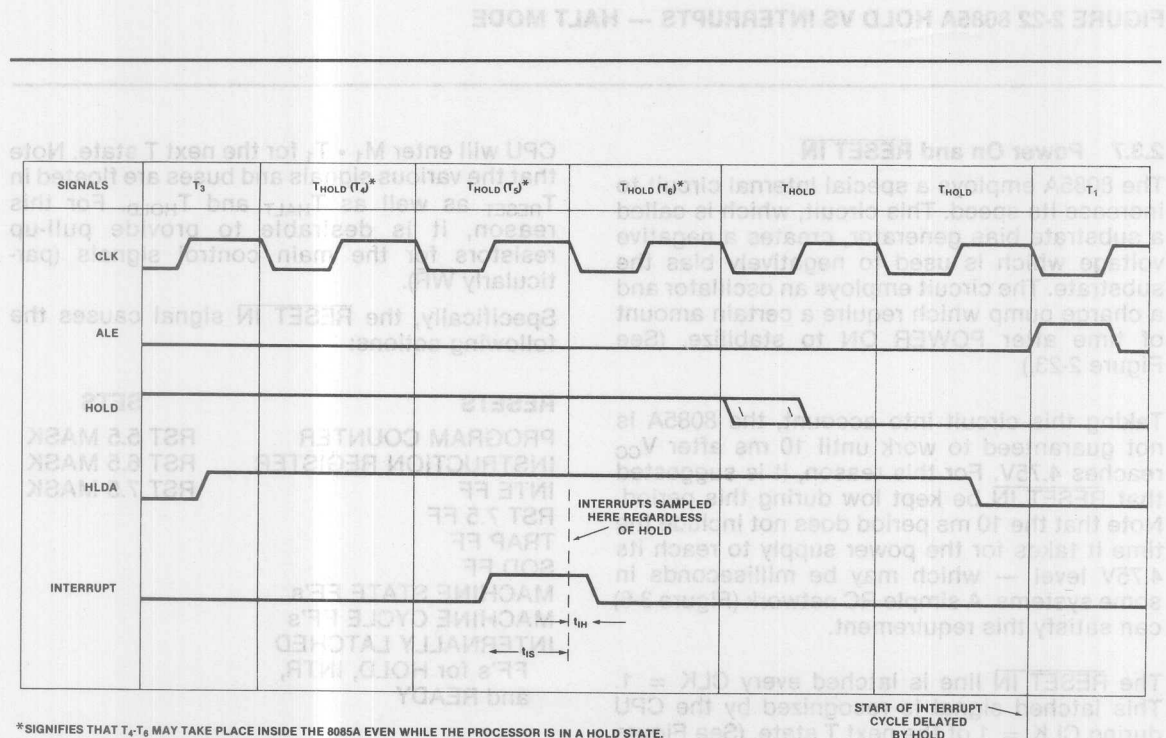
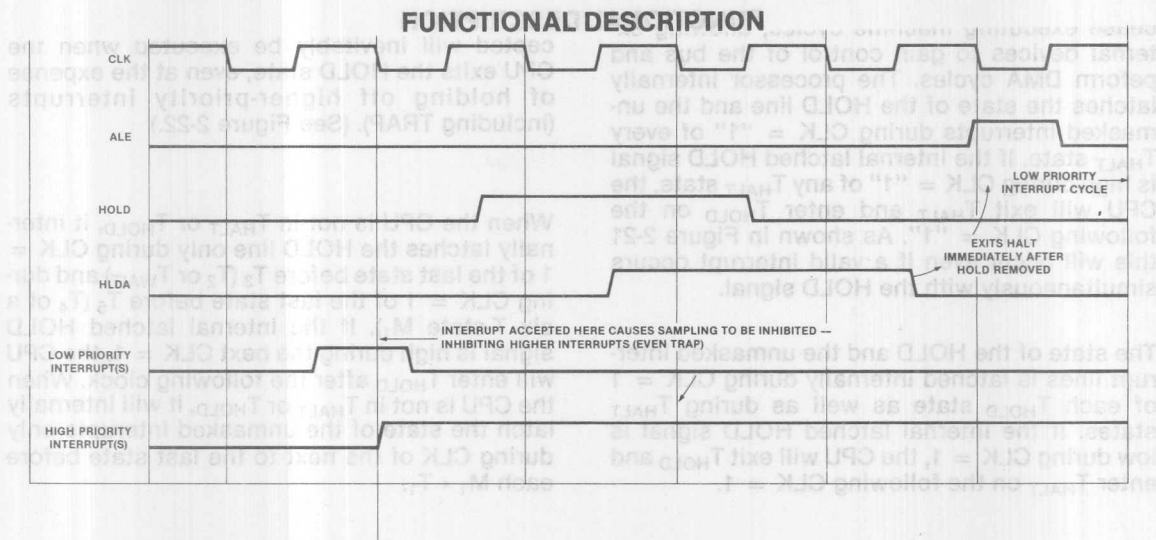


FIGURE 2-21 HOLD VS INTERRUPT — NON HALT



**FIGURE 2-22 8085A HOLD VS INTERRUPTS — HALT MODE**

### 2.3.7 Power On and RESET IN

The 8085A employs a special internal circuit to increase its speed. This circuit, which is called a substrate bias generator, creates a negative voltage which is used to negatively bias the substrate. The circuit employs an oscillator and a charge pump which require a certain amount of time after POWER ON to stabilize. (See Figure 2-23.)

Taking this circuit into account, the 8085A is not guaranteed to work until 10 ms after  $V_{CC}$  reaches 4.75V. For this reason, it is suggested that  $\overline{\text{RESET IN}}$  be kept low during this period. Note that the 10 ms period does not include the time it takes for the power supply to reach its 4.75V level — which may be milliseconds in some systems. A simple RC network (Figure 3-6) can satisfy this requirement.

The  $\overline{\text{RESET IN}}$  line is latched every  $\text{CLK} = 1$ . This latched signal is recognized by the CPU during  $\text{CLK} = 1$  of the next T state. (See Figure 2-24.) If it is low, the CPU will issue  $\overline{\text{RESET OUT}}$  and enter  $T_{\text{HALT}}$  for the next T state.  $\overline{\text{RESET IN}}$  should be kept low for a minimum of three clock periods to ensure proper synchronization of the CPU. When the  $\overline{\text{RESET IN}}$  signal goes high, the

CPU will enter  $M_1 \cdot T_1$  for the next T state. Note that the various signals and buses are floated in  $T_{\text{RESET}}$  as well as  $T_{\text{HALT}}$  and  $T_{\text{HOLD}}$ . For this reason, it is desirable to provide pull-up resistors for the main control signals (particularly  $\overline{\text{WR}}$ ).

Specifically, the  $\overline{\text{RESET IN}}$  signal causes the following actions:

#### RESETS

PROGRAM COUNTER  
INSTRUCTION REGISTER  
INTE FF  
RST 7.5 FF  
TRAP FF  
SOD FF  
MACHINE STATE FF's  
MACHINE CYCLE FF's  
INTERNALLY LATCHED  
FF's for HOLD, INTR,  
and READY

#### SETS

RST 5.5 MASK  
RST 6.5 MASK  
RST 7.5 MASK

$\overline{\text{RESET IN}}$  does not explicitly change the contents of the 8085A registers (A, B, C, D, E, H, L) and the condition flags, but due to  $\overline{\text{RESET IN}}$  occurring at a random time during instruction execution, the results are indeterminate.

## FUNCTIONAL DESCRIPTION

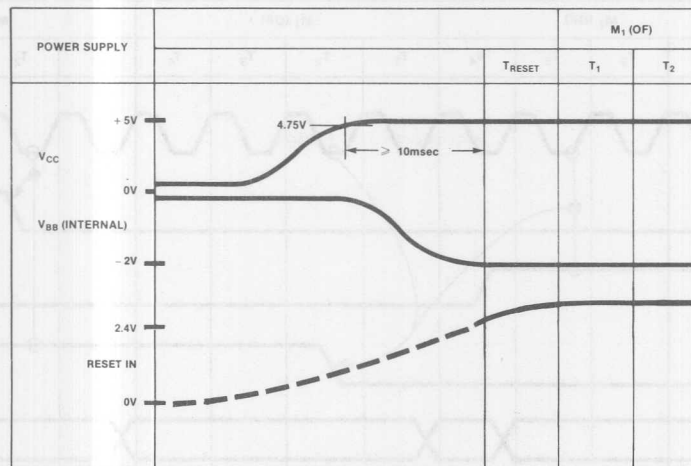


FIGURE 2-23 POWER-ON TIMING

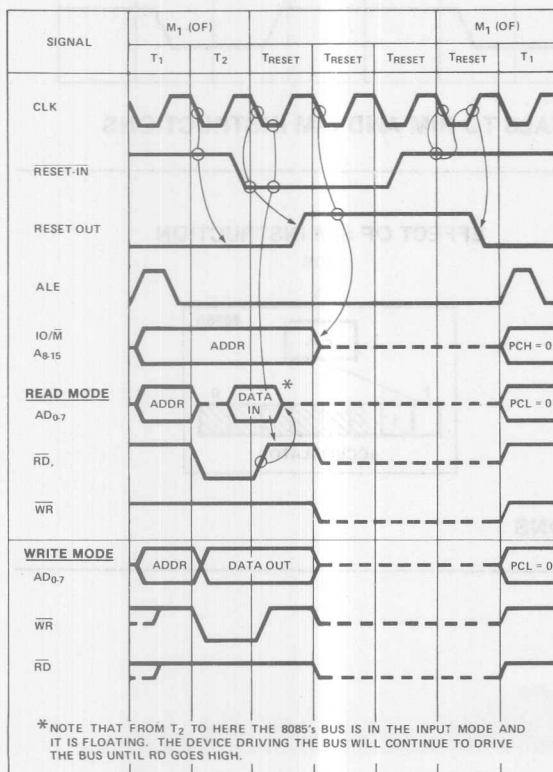


FIGURE 2-24 RESET IN TIMING

Following RESET, the 8085A will start executing instructions at location 0 with the interrupt system disabled, as shown in Figure 2-24.

Figure 2-24 also shows READ and WRITE operations being terminated by a RESET signal. Note that a RESET may prematurely terminate any READ or WRITE operation in process when the RESET occurs.

### 2.3.8 SID and SOD Signals:

Figure 2-25 shows the timing relationship of the SID and SOD signals to the RIM and SIM instructions. The 8085A has the ability to read the SID line into the accumulator bit 7 using RIM instructions. The state of the SID line is latched internally during T<sub>3</sub> • CLK = 0 of the RIM instruction. Following this, the state of the interrupt pins and masks are also transferred directly to the accumulator.

The 8085A can set the SOD flip-flop from bit 7 of the accumulator using the SIM instruction. (See Figure 2-26.) The data is transferred from the accumulator bit 7 to SOD during M<sub>1</sub> • T<sub>2</sub> • CLK = 0 of the instruction following SIM, assuming that accumulator bit 6 is a 1. Accumulator bit 6 is a "serial output enable" bit.

## FUNCTIONAL DESCRIPTION

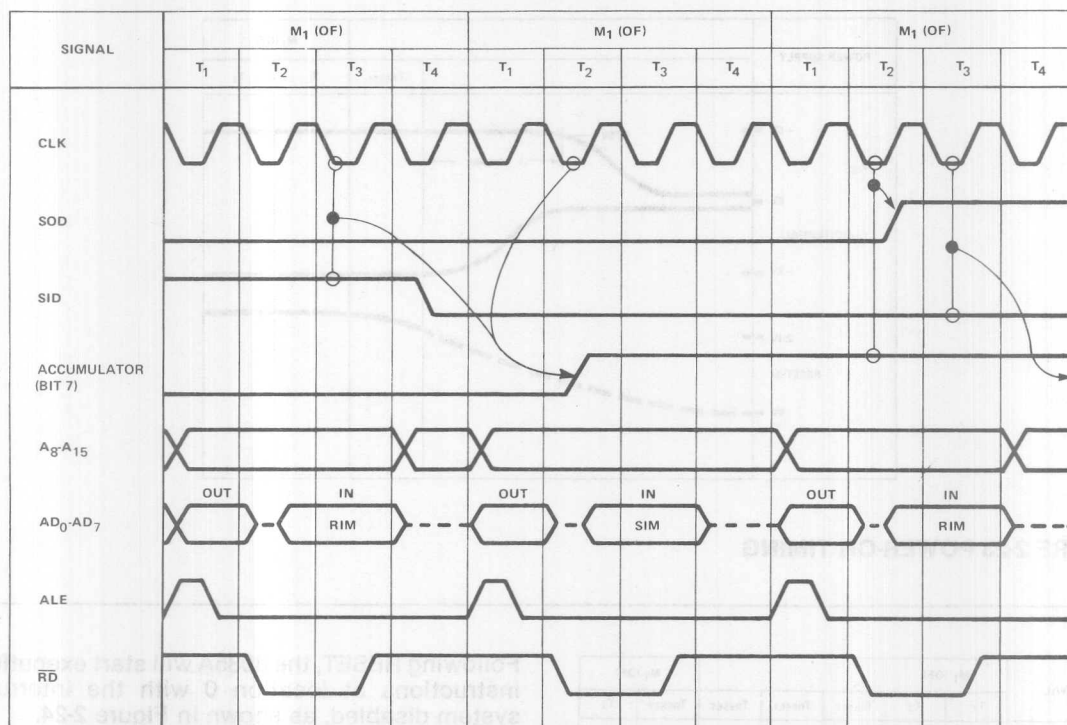
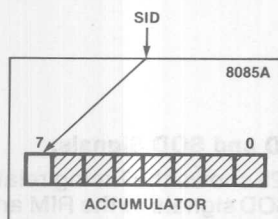


FIGURE 2-25 RELATIONSHIP OF SID AND SOD SIGNALS TO RIM AND SIM INSTRUCTIONS

### EFFECT OF RIM INSTRUCTION



### EFFECT OF SIM INSTRUCTION

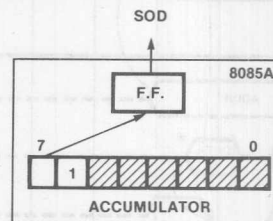


FIGURE 2-26 EFFECT OF RIM AND SIM INSTRUCTIONS

## 2.4 COMPARISON OF MCS-80 AND MCS-85 SYSTEM BUSES

This section compares the MCS-80 bus with the MCS-85 bus. Figure 2-28 details the signals and general timing of the two buses; the timing diagrams are drawn to the same scale (8080A clock cycle = 480 ns and 8085A clock cycle = 320 ns) to facilitate comparison.

## FUNCTIONAL DESCRIPTION

### MCS-80™ System Bus

The MCS-80 bus is terminated on one end by the CPU-GROUP (consisting of the 8080A, 8224, 8228) and on the other end by the various memory and I/O circuits. The following figure shows the major signals of the MCS-80 bus.

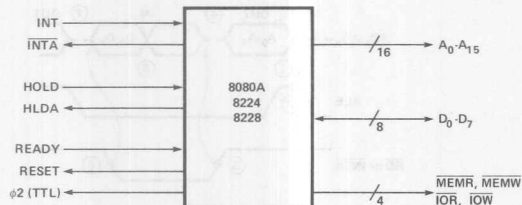


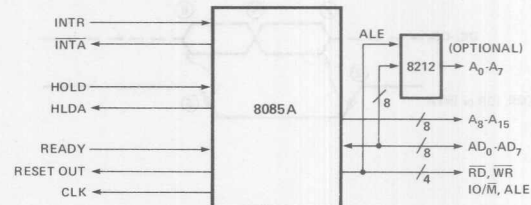
FIGURE 2-27 COMPARISON OF SYSTEM BUSES

### MCS-80™ System Bus

SIGNAL(S)	FUNCTION
A <sub>0</sub> -A <sub>15</sub>	The 16 lines of the address bus identify a memory or I/O location for a data transfer operation.
D <sub>0</sub> -D <sub>7</sub>	The 8 lines of the data bus are used for the parallel transfer of data between two devices.
MEMR, MEMW, IOR, IOW, INTA	These five control lines (MEMORY READ, MEMORY WRITE, I/O READ, I/O WRITE, and INTERRUPT ACKNOWLEDGE) identify the type and timing of a data transfer operation.
READY, RESET, HOLD, HLDA, φ2 (TTL), INT	These signals are used for the synchronization of slow speed memories, system reset, DMA, system timing, and CPU interrupt.

### MCS-85™ System Bus

The MCS-85 bus is terminated on one end by the 8085A and the other end by various memory and I/O devices. The MCS-85 bus may be optionally de-multiplexed with an 8212 eight bit latch to provide an MCS-80 type bus. The following figure shows the major signals of the MCS-85 bus.



### MCS-85™ System Bus

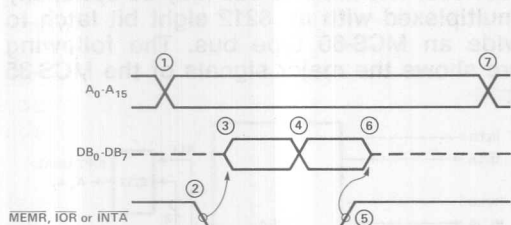
SIGNAL(S)	FUNCTION
A <sub>8</sub> -A <sub>15</sub>	These are the high order eight bits of the address, and are used to identify a memory or I/O location for a data transfer cycle.
AD <sub>0</sub> -AD <sub>7</sub>	These eight lines serve a dual function. During the beginning of a data transfer operation, these lines carry the low order eight bits of the address bus. During the remainder of the cycle, these lines are used for the parallel transfer of data between two devices.
RD, WR, INTA	These signals identify the type and timing of a data transfer cycle.
IO/M	The I/O/MEMORY line identifies a data transfer as being in the I/O address space or the memory address space.
ALE	ADDRESS LATCH ENABLE enables the latching of the A <sub>0</sub> -A <sub>7</sub> signals.
READY, RESET OUT, HOLD, HLDA, CLK, INTR	These signals are used for the synchronization of slow speed memories, system reset, DMA, system timing and CPU interrupt.

FIGURE 2-28 COMPARISON OF SYSTEM BUSES

## FUNCTIONAL DESCRIPTION

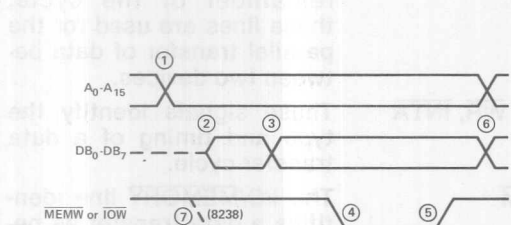
## MCS-80™ System Bus

The basic timing of the MCS-80 BUS for a READ CYCLE is as follows:



The MCS-80 first presents the address ① and shortly thereafter the control signal ②. The data bus, which was in the high impedance state, is driven by the selected device ③. The selected device eventually presents the valid data to the processor ④. The processor raises the control signal ⑤, which causes the selected device to put the data bus in the high impedance state ⑥. The processor then changes the address ⑦ for the start of the next data transfer.

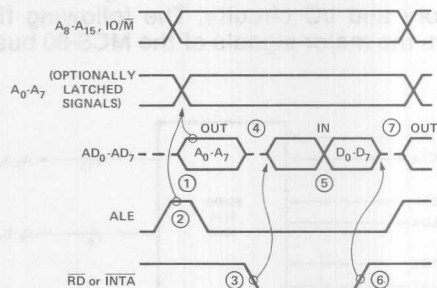
The basic timing of the MCS-80 BUS for a WRITE CYCLE is as follows:



The MCS-80 first presents the address ①, then enables the data bus driver ②, and later presents the data ③. Shortly thereafter, the MCS-80 drops the control signal ④ for an interval of time and then raises the signal ⑤. The MCS-80 then changes the address ⑥ in preparation for the next data transfer. The advance write signal of the 8238 is also shown ⑦.

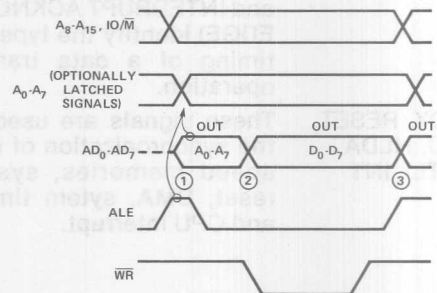
## MCS-85™ System Bus

The basic timing of the MCS-85 BUS for a READ CYCLE is as follows:



At the beginning of the READ cycle, the 8085A sends out all 16 bits of address ①. This is followed by ALE ② which causes the lower eight bits of address to be latched in either the 8155/56, 8355, 8755A, or in an external 8212.  $\overline{RD}$  is then dropped ③ by the 8085A. The data bus is then tri-stated by the 8085A in preparation for the selected device driving the bus ④; the selected device will continue to drive the bus with valid data ⑤, until  $\overline{RD}$  is raised ⑥ by the 8085A. At the end of the READ CYCLE ⑦, the address and data lines are changed in preparation for the next cycle.

The basic timing of the MCS-85 BUS for a WRITE CYCLE is as follows:



The timing of the WRITE CYCLE is identical to the MCS-85 READ CYCLE with the exception of the AD<sub>0</sub>-AD<sub>7</sub> lines. At the beginning of the cycle ①, the low order eight bits of address are on AD<sub>0</sub>-AD<sub>7</sub>. After ALE drops, the eight bits of data ② are put on AD<sub>0</sub>-AD<sub>7</sub>. They are removed ③ at the end of the WRITE CYCLE, in anticipation of the next data transfer.

**FIGURE 2-28 (Continued) COMPARISON OF SYSTEM BUSES**

The following observations of the two buses can be made:

1. The access times from address leaving the processor to returning data are almost identical, even though the 8085A is operating 50% faster than the 8080.
2. With the addition of an 8212 latch to the 8085A, the basic timings of the two systems are very similar.
3. The 8085A has more time for address setup to  $\overline{RD}$  than the 8080.
4. The MCS-80 has a wider  $\overline{RD}$  signal, but a narrower  $\overline{WR}$  signal than the 8085A.
5. The MCS-80 provides stable data setup to the leading and trailing edges of  $\overline{WR}$ , while the 8085 provides stable data setup to only the trailing edge of  $\overline{WR}$ .
6. The MCS-80 control signals have different widths and occur at different points in the machine cycle, while the 8085A control signals have identical timing.
7. While not shown on the chart, the MCS-80 data and address hold times are adversely affected by the processor preparing to enter the HOLD state. The 8085A has identical timing regardless of entering HOLD.
8. Also not shown on the chart is the fact that all output signals of the 8085A have  $-400\mu\text{A}$  of source current and 2.0 ma of sink current. The 8085A also has input voltage levels of  $V_{IL} = 0.8\text{V}$  and  $V_{IH} = 2.0\text{V}$ .

## CONCLUSION:

The preceding discussion has clearly shown that the MCS-85 bus satisfies the two restrictions of COMPATIBILITY and SPEED. It is compatible because it requires only an 8212 latch to generate an MCS-80 type bus. If the four control signals  $\overline{MEMR}$ ,  $\overline{MEMW}$ ,  $\overline{IOR}$  and  $\overline{IOW}$  are desired, they can be generated from  $\overline{RD}$ ,  $\overline{WR}$ ,

and  $\overline{IO/\overline{M}}$  with a decoder or a few gates. The MCS-85 bus is also fast. While running at 3MHz, the 8085A generates better timing signals than the MCS-80 does at 2MHz. Furthermore, the multiplexed bus structure doesn't slow the 8085A down, because it is using the internal states to overlap the fetch and execution portions of different machine cycles. Finally, the MCS-85 can be slowed down or sped up considerably, while still providing reasonable timing.

TO USE. The  $\overline{RD}$ ,  $\overline{WR}$ , and  $\overline{INTA}$  control signals all have identical timing, which isn't affected by the CPU preparing to enter the HOLD state. Furthermore, the address and data bus have good setup and hold times relative to the control signals. The voltage and current levels for the interface signals will all drive buses of up to 40 MOS devices, or 1 schottky TTL device.

The MCS-85 system bus is also EFFICIENT. Efficiency is the reason that the lower eight address lines are multiplexed with the data bus. Every chip that needs to use both  $A_0-A_7$  and  $D_0-D_7$  saves 7 pins (the eighth pin is used for ALE) on the interface to the processor. That means that 7 more pins per part are available to either add features to the part or to use a smaller package in some cases. In the three chip system shown in Figure 3-6, the use of the MCS-85 bus saves  $3 \times 7 = 21$  pins, which are used for extra I/O and interrupt lines. A further advantage of the MCS-85 bus is apparent in Figure 3-7, which shows a printed circuit layout of the circuit in Figure 3-6. The reduced number of pins and the fact that compatible pinouts were used, provides for an extremely compact, simple, and efficient printed circuit. Notice that great care was taken when the pinouts were assigned to ensure that the signals would flow easily from chip to chip to chip.



## CHAPTER 3

### System Operation and Interfacing





## CHAPTER 3 SYSTEM OPERATION AND INTERFACING

### 3.1 INTERFACING TO THE 8085A

The 8085A interfaces to both memory and I/O devices by means of READ and WRITE machine cycles, the timing of which are identical. During each machine cycle the 8085A issues an address and a control signal, then either sends data out on the bus or reads data from the bus. The 8085A may be performing a READ machine cycle, but what it reads could be a ROM, RAM, I/O device, peripheral device, or nothing.

There is no distinction between data, instruction opcodes, and I/O port numbers except the way the CPU interprets what it reads from the bus. If an opcode is what would logically appear on the bus, the CPU will treat as an opcode whatever does appear there; if an I/O port number is to be expected, what appears will be interpreted as a port number. The same is true for a WRITE cycle. The 8085A issues an address, data, and a control signal. Unless it is requested to WAIT (by use of the READY line) it will complete the cycle and proceed to the next. Regardless of whether there is a device present to accept the data, the CPU executes one instruction at a time, in sequence, until told to do otherwise. The program controls the sequence and nature of all machine cycles until an interrupt occurs.

There are two ways of addressing I/O devices in the MCS-85 system. If the  $\text{IO}/\overline{\text{M}}$  output from the CPU is used to distinguish between I/O and memory READ and WRITE cycles, then that system is said to employ standard, or I/O-mapped, I/O. If  $\text{IO}/\overline{\text{M}}$  is not so used, the CPU does not distinguish between I/O and memory, and its system employs memory-mapped I/O. Each method of addressing I/O has advantages and disadvantages.

### 3.2 MEMORY-MAPPED I/O

#### 3.2.1 Advantages of Memory-Mapped I/O

Since the processor doesn't distinguish I/O from memory using this addressing scheme, you can take advantage of the larger instruction set that references the memory address space. Instead of only being able to transfer a byte of data between the accumulator and the I/O port (using INPUT and OUTPUT instructions), you can now program

arithmetic and logic operations on port data as well as move data between any internal register and the I/O port. Consider the new meaning of the following instructions:

Examples:

MOVr,M	(Input Port to any Register)
MOV M,r	(Output any Register to Port)
MVI M	(Output immediate data to Port)
LDA	(Input Port to ACC)
STA	(Output from ACC to Port)
LHLD	(16-Bit Input)
SHLD	(16-Bit Output)
ADD M	(Add Port to ACC)
ANA M	(AND Port with ACC)

#### 3.2.2 Disadvantages of Memory-Mapped I/O

While memory instructions may increase the flexibility of the I/O system, there are some drawbacks. Since I/O devices are now addressed as memory, there are fewer addresses available for memory. A common practice is to use address bit 15 ( $A_{15}$ ) to distinguish memory from I/O. (See Figure 3-2 and accompanying discussion.) If  $A_{15} = 0$  then memory is being addressed; if  $A_{15} = 1$ , I/O is being addressed. This particular scheme limits the maximum amount of memory that can be used to 32k bytes. A further disadvantage of memory-mapped I/O is that it takes 3 bytes of instruction and 13 clock cycles using the LDA or STA instructions to specify moving a byte of data between the accumulator and an I/O device, whereas the INPUT and OUTPUT instructions require only two bytes and 10 clock cycles. This is because the I/O address space is smaller (only 256 bytes) and therefore requires fewer bits to completely specify an address. A further advantage of using the INPUT and OUTPUT Instructions is that it allows the easy connection of the MCS-80 peripherals to the MCS-85 multiplexed bus. If you memory-map the MCS-80 peripherals to the MCS-85 bus, you must either latch the lower address bits with an 8212 or use a portion of the memory address space by connecting the chip selects and address lines of the ports to the unmultiplexed upper eight lines of the address bus.

### 3.3 ADDRESS ASSIGNMENT

#### 3.3.1 Decoding

Besides memory-mapped I/O, another practice is to only partially decode the address bus when generating chip selects. Every device has a given number of unique addresses associated with it. The 8355, for instance, has 2k bytes of ROM and therefore has 2k addresses associated with the ROM. Any one of these 2k addresses can be uniquely specified by a pattern on the 11 ( $2^{11} = 2k$ ) address lines. However, since the 8355 must work with other devices in a system, it isn't enough to simply specify the 11 bits; further bits of information must be used to locate the 2k bytes within the 65k address space. The 2k bytes are located by the use of chip enable (CE) inputs to the 8355 chip. If the 8355 were to occupy the first 2k bytes of the memory address space, it would, strictly speaking, be necessary to decode the fact that  $A_{15}-A_{11}$  were all zeroes, and use that condition as a chip enable. Then the 8355 would be selected only when the address bus was less than 2k.

However, if other 2k blocks of addresses aren't being used, you may combine those addresses and not decode all of the upper five address lines for chip enables. In fact, in a small system you may need to decode only one bit of address, which is to say connect that bit of the address bus to the chip enable line of the 8355. If you connect  $A_{11}$  to the  $\overline{CE}$  line of the 8355 and tie CE to  $V_{CC}$ , then the 8355 would be selected whenever the memory address was less than 2k. (See Figure 3-1A.)

However, it will also be selected whenever memory locations 4k-6k, 8k-10k, 61k-63k (i.e., whenever bit  $A_{11} = 0$ ) is addressed. If the programmer is aware of this, and if there are no other devices assigned to the other address spaces, then it may be an acceptable condition. Care must be taken, however, to ensure that at no time will two different devices be selected simultaneously. Whenever one device is selected, that memory address must deselect all other devices. If two devices are selected simultaneously for a READ operation, the electrical conflict on the bus may damage one or both parts. Note also that the address bus may reflect an undesired address during  $T_5$ ,  $T_6$  of an opcode fetch cycle and during address bus transitional periods in  $T_1$  (this is illustrated in Chapter 2). Therefore, all memory and I/O devices must qualify their selection with  $\overline{RD}$  or  $\overline{WR}$ , or the address on the bus at the falling edge of the ALE, so as to ignore all spurious addresses.

#### 3.3.2 Linear Selection

Using an address bit as a chip select is referred to as linear selection. The direct consequence of linear selection is that you cut the available address space in half for each single address bit used as a chip enable. If this penalty is too high, you can always use an 8205 one-of-eight decoder. Also, some chips have multiple chip enables, which allows for some automatic decoding of the address. (See Figures 3-1B and 3-1C.)

One drawback to linear selection is that the memory addresses of the different parts are not contiguous. For example, if three 8355s are addressed using linear selection, one might be located at 0-2k, the next at 6k-8k, and the next at 10k-12k. The programmer must recognize these page boundaries and jump over them.

### 3.4 INTERFACING TO THE 8155/8156, 8355/8755A

#### 3.4.1 I/O Mapped I/O:

This section describes some of the techniques involved in connecting the MCS-85 combination memory and I/O chips to the 8085A as I/O devices.

Figure 3.1A shows one 8355 connected to the 8085A bus. (In the interest of simplicity, only the chip enable and  $IO/\overline{M}$  lines are shown; the other lines are connected as shown in Figures 3.6, 3.7 or 3.8.) Notice that CE is tied to  $V_{CC}$  and  $\overline{CE}$  is connected to  $A_{11}$ . This is because after RESET the processor always starts executing at location 0. Since the ROM normally contains the program, it must be selected when the address is all zeroes.

One consequence of the ROM being selected by an all-zero address is that the I/O ports on the chip will be selected only when  $A_{11} = 0$ . This is because the I/O ports and the memory have common chip enables, therefore forcing the selection conditions of one onto the other. Furthermore, since the  $IO/\overline{M}$  line of the chip is connected to the  $IO/\overline{M}$  line of the 8085A, the port has I/O mapped I/O. The I/O ports can be accessed only by use of the INPUT and OUTPUT instructions; since these are the only instructions that cause  $IO/\overline{M}$  to go high.

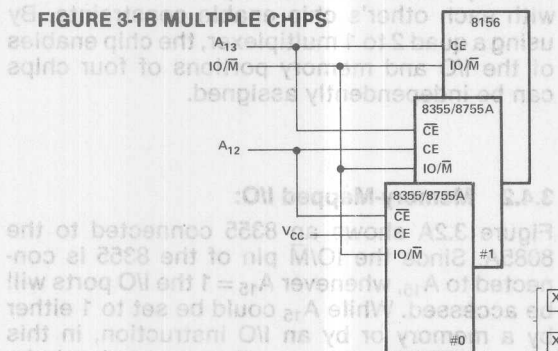
The boxes to the right of the chip in Figure 3.1A indicate the memory addresses and I/O Port numbers required to access the chip. As a result of the linear selection technique used, there are many "don't care" bits (marked by "X's") in the address. While they don't affect the addressing of this device, they may affect other

## SYSTEM OPERATION

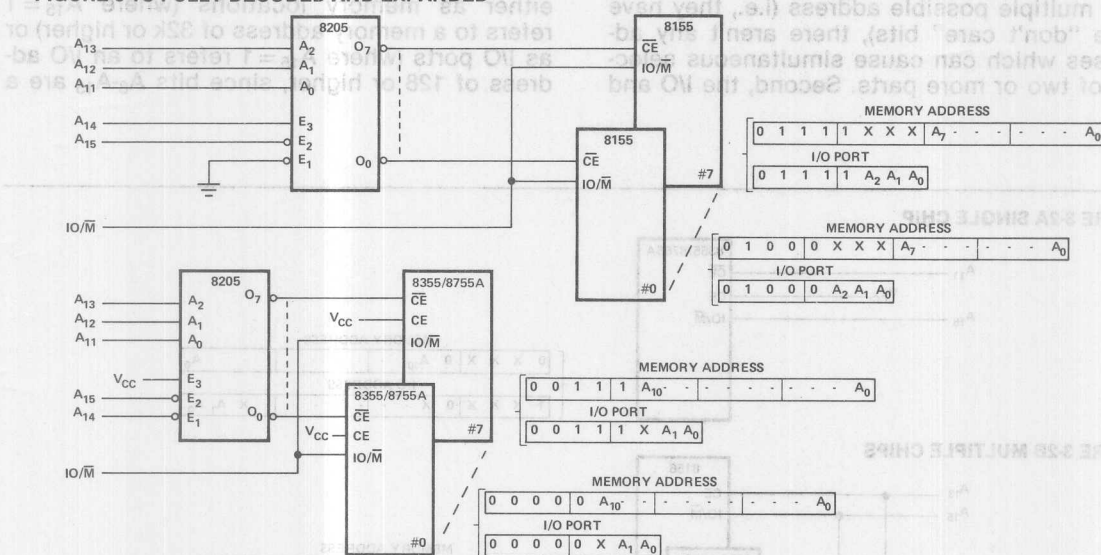
**FIGURE 3-1A SINGLE CHIP**



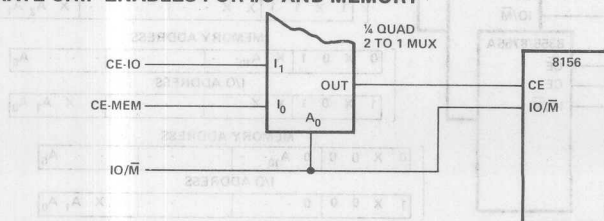
**FIGURE 3-1B MULTIPLE CHIPS**



**FIGURE 3-1C FULLY DECODED AND EXPANDED**



**FIGURE 3-1D SEPARATE CHIP ENABLES FOR I/O AND MEMORY**



**FIGURE 3-1 MCS-85™ PERIPHERALS WITH I/O MAPPED I/O**

devices may not be selected simultaneously, thus each device must have an address that not only selects itself, but also deselects all other devices. If there are any bits which are truly "don't cares," they are customarily assigned to be zero. If all the "X" bits in Figure 3.1A were "don't cares," then the chip could be addressed as memory locations 0-2k, and I/O Ports 0-3.

Figure 3.1B shows a slightly larger system of two 8355s and one 8156. Notice that 8355 No. 1 uses its two chip enable lines to decode  $A_{12} = 1$ ,  $A_{11} = 0$ . It is possible to address each of the chips without selecting any of the others. Also notice that there are some illegal addresses (e.g.,  $A_{12} = 0$ ,  $A_{11} = 1$ ) that would cause two of the devices to turn on simultaneously. The programmer must not use these addresses.

Figure 3.1C shows a larger MCS-85 system. Two 8205s are used to completely decode the addresses. There are some interesting points to observe here. First, while some of the devices have multiple possible address (i.e., they have some "don't care" bits), there aren't any addresses which can cause simultaneous selection of two or more parts. Second, the I/O and

each time, constraints. Third, only one 8205 is required per eight chips for the decoding; that's an overhead of only 1/8 of a chip per part.

Figure 3.1D shows a remedy to the problem illustrated in Figure 3.1C, namely that I/O and memory portions of the chip are forced to live with each other's chip enable constraints. By using a quad 2 to 1 multiplexer, the chip enables of the I/O and memory portions of four chips can be independently assigned.

### 3.4.2 Memory-Mapped I/O:

Figure 3.2A shows an 8355 connected to the 8085A. Since the IO/M pin of the 8355 is connected to  $A_{15}$ , whenever  $A_{15} = 1$  the I/O ports will be accessed. While  $A_{15}$  could be set to 1 either by a memory or by an I/O instruction, in this situation the port is usually accessed only by the memory instructions. You may access ports either as memory locations (where  $A_{15} = 1$  refers to a memory address of 32k or higher) or as I/O ports (where  $A_{15} = 1$  refers to an I/O address of 128 or higher, since bits  $A_8$ - $A_{15}$  are a

FIGURE 3-2A SINGLE CHIP

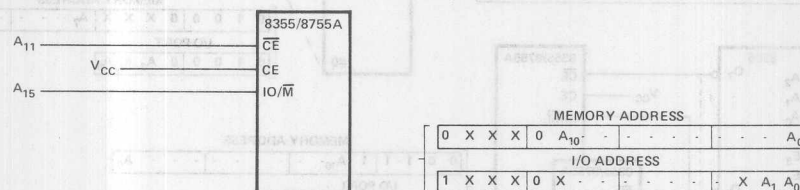


FIGURE 3-2B MULTIPLE CHIPS

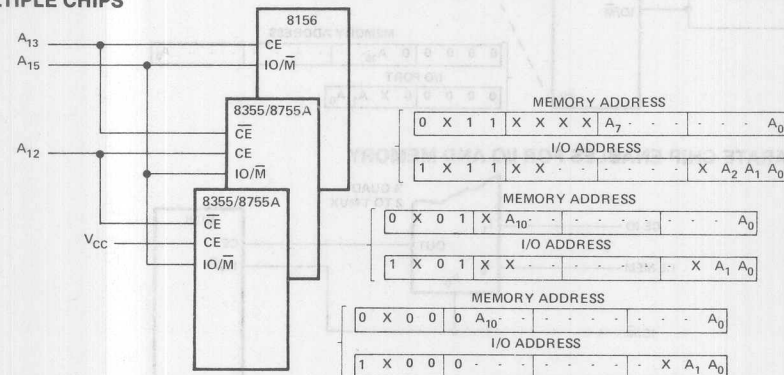


FIGURE 3-2 MCS-85™ PERIPHERALS WITH MEMORY-MAPPED I/O

replication of bits  $A_0-A_7$ ). Assuming that memory-mapped I/O is used, the addresses are shown in the boxes to the right in Figure 3-2. If you want to be sure that neither the I/O nor the memory is ever selected by any INPUT or OUTPUT instruction, then the chip enable must be conditioned by  $IO/\overline{M} = 0$ .

Figure 3.2B shows a somewhat larger system, also using memory-mapped I/O. As in Figure 3.1B care must be exercised to ensure that no two devices are accessed simultaneously. You can see that considerable memory address space is used up as a result of using memory-mapped I/O.

## 3.5 INTERFACING TO MCS-80™ PERIPHERALS

### 3.5.1 I/O Mapped I/O:

For want of a better name, the Intel® 825x, 827x, and 829x series peripherals are referred to here as MCS-80 peripherals because unlike the 8155/56, 8355 and 8755A, they are compatible with the nonmultiplexed MCS-80 system bus.

To interface to an MCS-80 peripheral, you must provide a constant address, a chip select, and  $\overline{RD}$  or  $\overline{WR}$ . Since the upper address lines ( $A_8-A_{15}$ ) of the 8085A are nonmultiplexed, they can be tied directly to the peripherals, as shown in Figure 3.3A. To provide I/O mapped I/O, use either linear selection (keeping the I/O and memory addresses noncoincident), or condition the chip selects  $\overline{WR}$  with  $IO/\overline{M} = 1$ . Figure 3.3A shows a technique of gating the chip selects with  $IO/\overline{M} = 1$ , using an 8205. This technique also allows more I/O devices to be used than linear selection would. Note that this technique relies on the fact that the I/O Port number is copied onto  $A_8-A_{15}$  as well as  $A_0-A_7$  during an INPUT or OUTPUT instruction.

Figure 3.3B shows an alternative approach to interfacing to MCS-80 components. By latching the lower 8 bits of address with an 8212, and decoding the control signals with an 8205, you create an exact copy of the MCS-80 (8080A, 8224, 8228) bus. You may then use whatever circuits have been previously developed for the 8080. The total cost is one 8212 and one 8205. Since the same signals might have needed buffering anyway (and the 8212 and 8205 provide buffering of their outputs), the extra component overhead ranges from little to nothing.

### 3.5.2 Memory-Mapped I/O:

Exactly the same techniques used to memory map the MCS-85 apply to the MCS-80 I/O devices. Figure 3.4 shows an 8205 used to qualify the chip select of the I/O device with  $IO/\overline{M} = 0$ . Since

the MCS-80 peripherals require nonmultiplexed address lines, linear select is not too useful unless the address lines are latched. This is because connecting both the chip selects and the address lines of the MCS-80 peripherals to  $A_8-A_{15}$  would deplete all the useful addresses very quickly.

## 3.6 INTERFACING TO STANDARD BUS MEMORIES

Standard bus memory devices are designed to be used with nonmultiplexed address and data buses. Interfacing to standard memories is very similar to interfacing to MCS-85 memories with the exception that  $A_0-A_7$  must be latched. Once this requirement is met, all the tricks discussed earlier can be used. Since the address lines would eventually require buffering as the system size grew, the overhead of the 8212 latch again becomes negligible.

Figure 3.5 shows the interface of the 8085A to a large block of memory, specifically 16k bytes of ROM and 8k bytes of RAM. Besides the memories, the circuit requires only 2-1/6 other parts for logical gating. If MCS-80 I/O parts were used, the 8212 latch could be shared between the two groups, further reducing the gating overhead per IC. Sixteen 2142 chips and eight 2316E chips are used in this design. The data bus, address lines 8-10, and control signals in this system all should be buffered. This applies to any system with the number of memory devices represented here.

Wherever two or more parts are paralleled on the same bus, they must be 3-state devices such as the 2142 RAM, 2316E ROM, 2716 EPROM, 2332 ROM, 2732 EPROM, and 2364 ROM, which have either an output disable (OD) input or multiple chip select (CS) inputs. To prevent bus contention, only one memory device may be output-enabled at a time in this configuration; the outputs of all others must be deselected during  $\overline{RD}$ .

For additional information on interfacing standard memory devices, please read Section 2 of Appendix I and the Intel applications note AP-30 "Application of Intel's 5V EPROM and ROM Family for Microprocessor Systems" available from: Intel, Literature Dept., 3065 Bowers Ave., Santa Clara, CA 95051.

## 3.7 DYNAMIC RAM INTERFACE:

For interfacing the dynamic RAM, Intel makes a single-component dynamic RAM refresh controller, the 8202, which interfaces the 8085A to multiplexed-address-bus dynamic RAMs like

## SYSTEM OPERATION

FIGURE 3-3A DECODED CHIP SELECTS

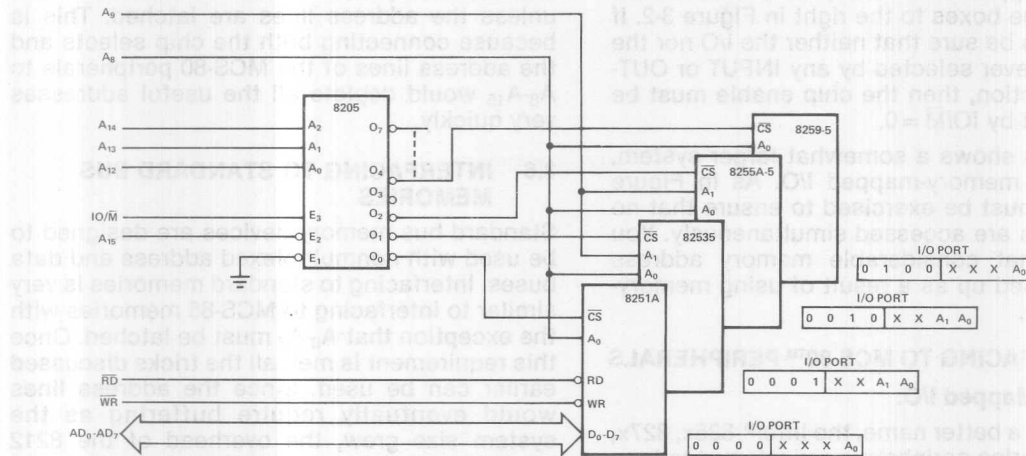


FIGURE 3-3B DECODED CONTROLS AND LATCHED ADDRESS (MCS-80™ TYPE BUS)

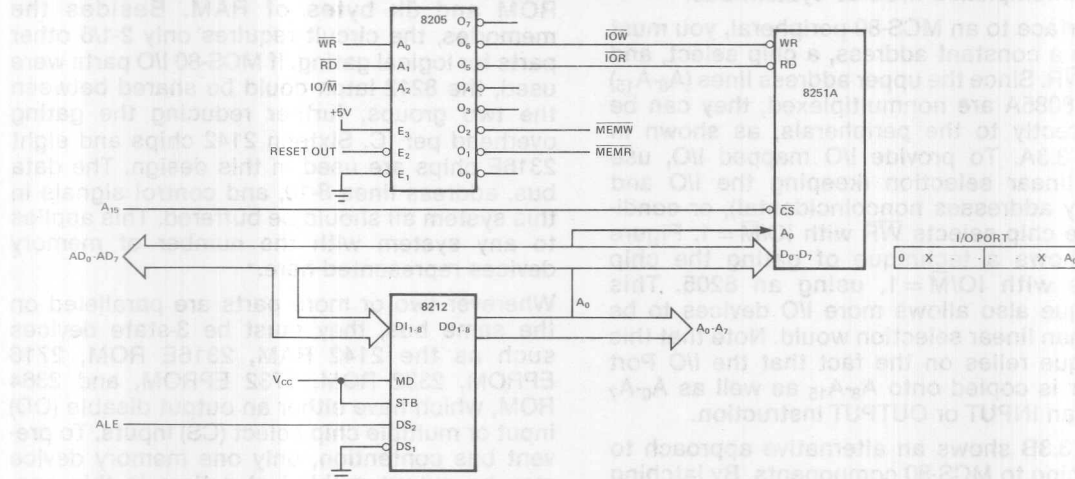


FIGURE 3-3 MCS-80™ PERIPHERALS WITH I/O MAPPED I/O

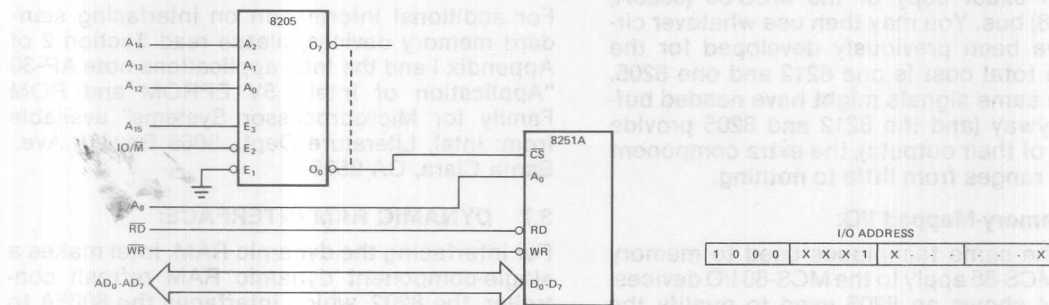


FIGURE 3-4 MCS-80™ PERIPHERALS WITH MEMORY-MAPPED I/O AND DECODED CHIP SELECTS

## SYSTEM OPERATION

the Intel 2104A and 2117. The 8202 provides the necessary refreshing for such dynamic RAMs, and also provides the control signals required for accessing, selecting, and address clocking. It allows for the use of the 8085A's full capability of 64k bytes of address space with no additional buffering devices. As with other standard memory interfaces, it is necessary to demultiplex the lower 8 bits of address from the multiplexed 8085A bus, AD<sub>0-7</sub>.

### 3.8 MINIMUM MCS-85™ SYSTEM

The Schematics of Figure 3.6 depict a minimum system core. In actual use, some of the processor control signals (TRAP, INTR, and HOLD) would have to be terminated. Also, interface logic to external devices as well as more memory and I/O devices may be desirable. The first thing one notices about the system in Figure 3.6 is the scarcity of parts required to build this system. With a minimum of parts, we

have constructed a microcomputer system that has the following functions:

PARTS	FUNCTIONS
1 8085A	1 CPU (Clock cycle ≤ 320 ns)
1 8355/8755A	2048 Bytes of either EPROM or ROM
1 8156	256 Bytes of RAM
1 Crystal	38 I/O Lines
4 Resistors	5 Interrupts
1 Capacitor	1 Programmable Timer/Counter
1 Diode	1 Crystal and Oscillator
1 + 5 Power Supply	1 Clock
	1 Power-on Reset

By looking at the printed circuit layout of Figure 3.7, we can see that not only are there just 3 ICs, but that the interconnection of these parts is extremely easy and provides a very dense layout. Especially notice the easy flow of the system bus on the solder side of the board.

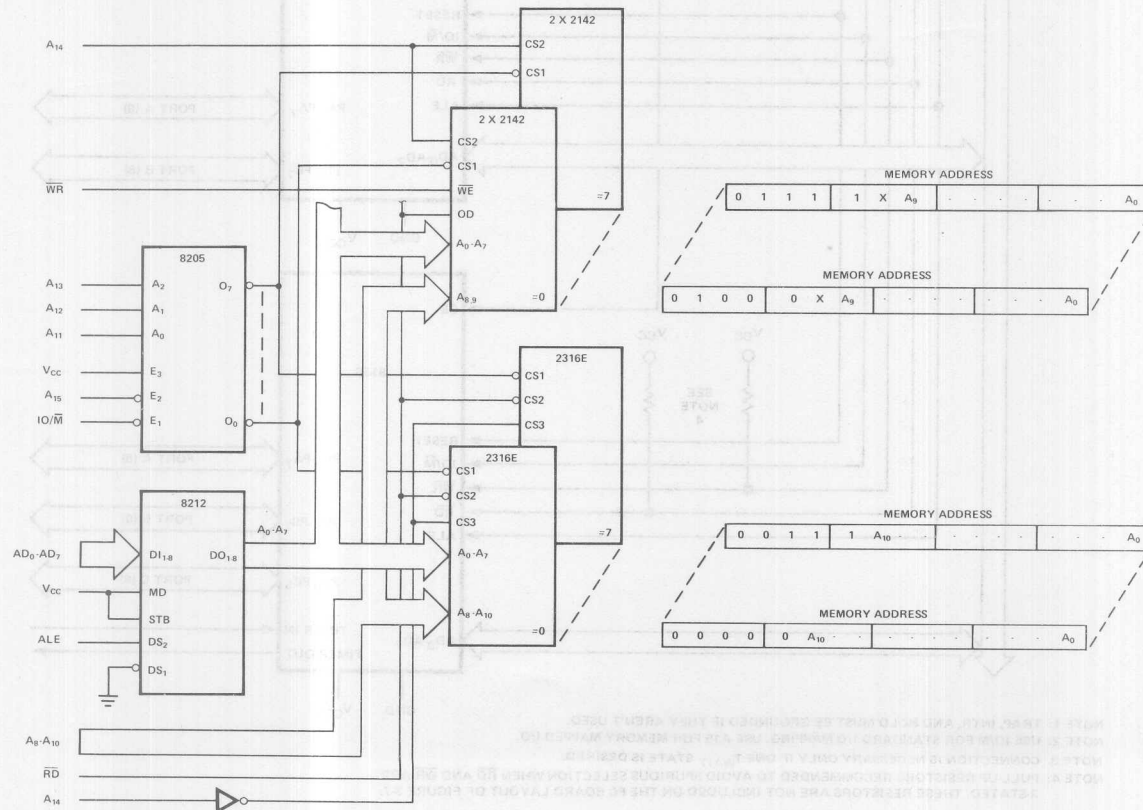
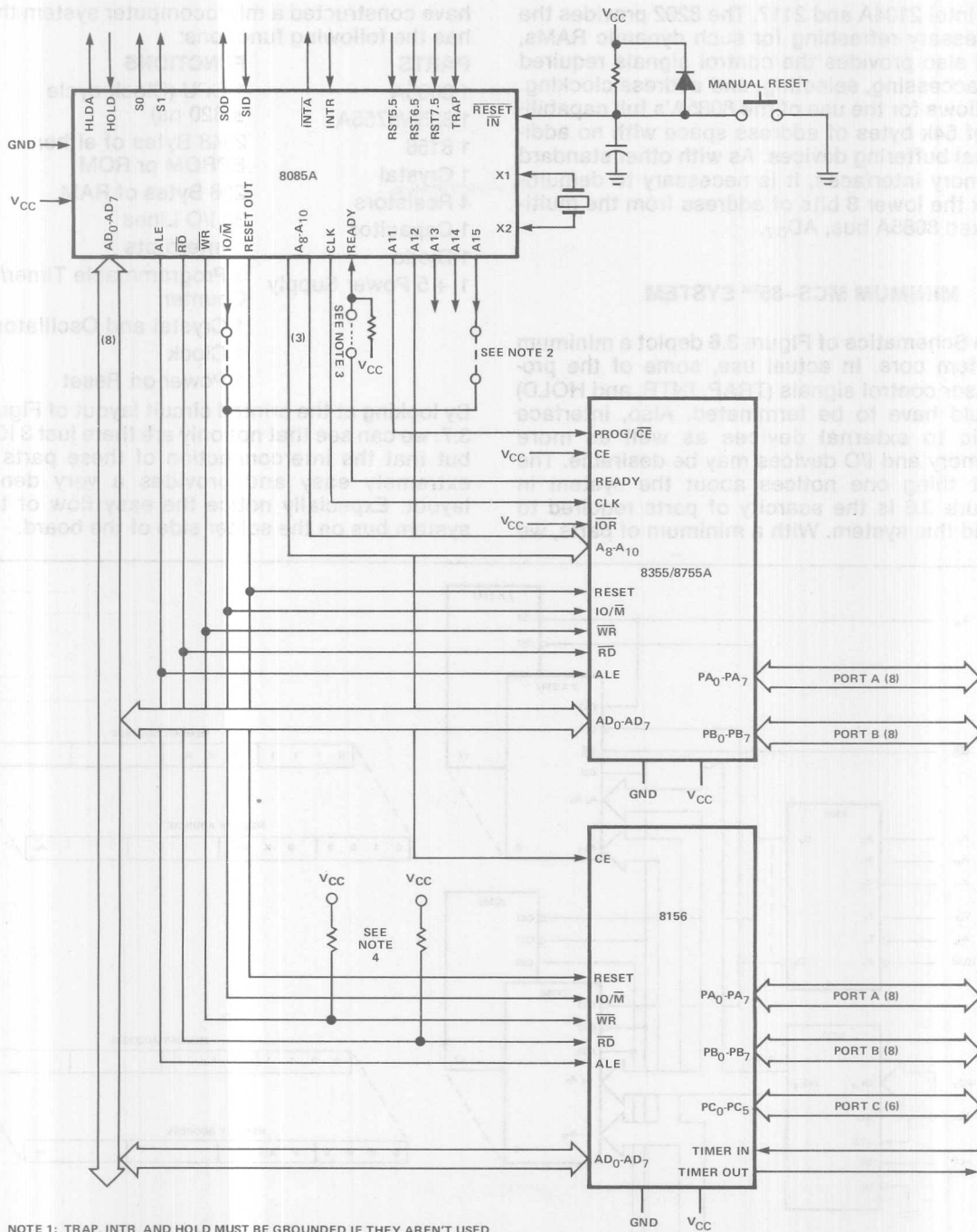


FIGURE 3-5 STANDARD MEMORIES WITH LATCHED ADDRESS AND DECODED CHIP SELECTS

## SYSTEM OPERATION



- NOTE 1: TRAP, INTR, AND HOLD MUST BE GROUNDLED IF THEY AREN'T USED.  
 NOTE 2: USE IO/M FOR STANDARD I/O MAPPING. USE A15 FOR MEMORY MAPPED I/O.  
 NOTE 3: CONNECTION IS NECESSARY ONLY IF ONE  $T_{WAIT}$  STATE IS DESIRED.  
 NOTE 4: PULL-UP RESISTORS RECOMMENDED TO AVOID SPURIOUS SELECTION WHEN  $\overline{RD}$  AND  $\overline{WR}$  ARE 3-STATE. THESE RESISTORS ARE NOT INCLUDED ON THE PC BOARD LAYOUT OF FIGURE 3-7.

FIGURE 3-6 MINIMUM 8085 SYSTEM

# SYSTEM OPERATION

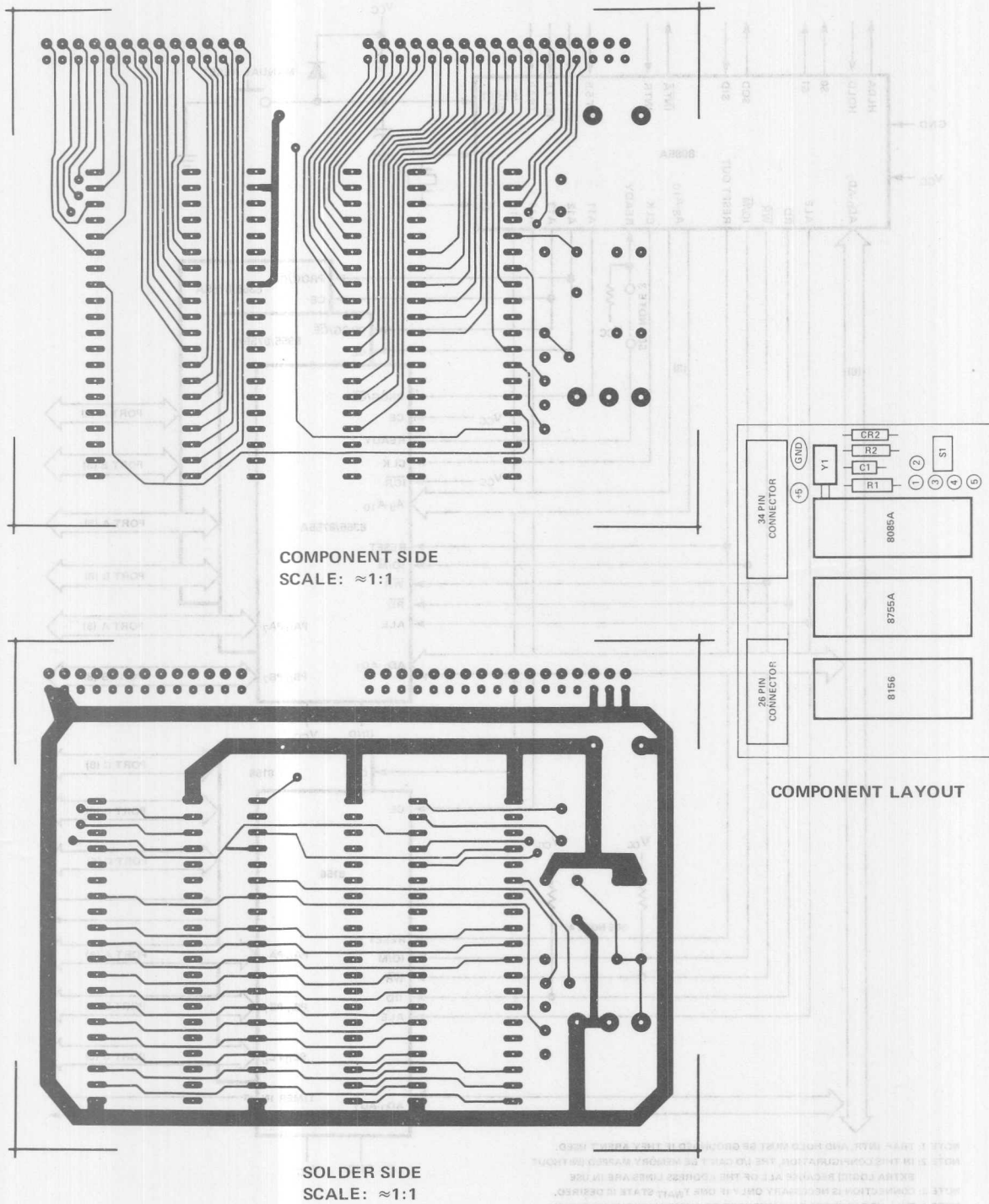
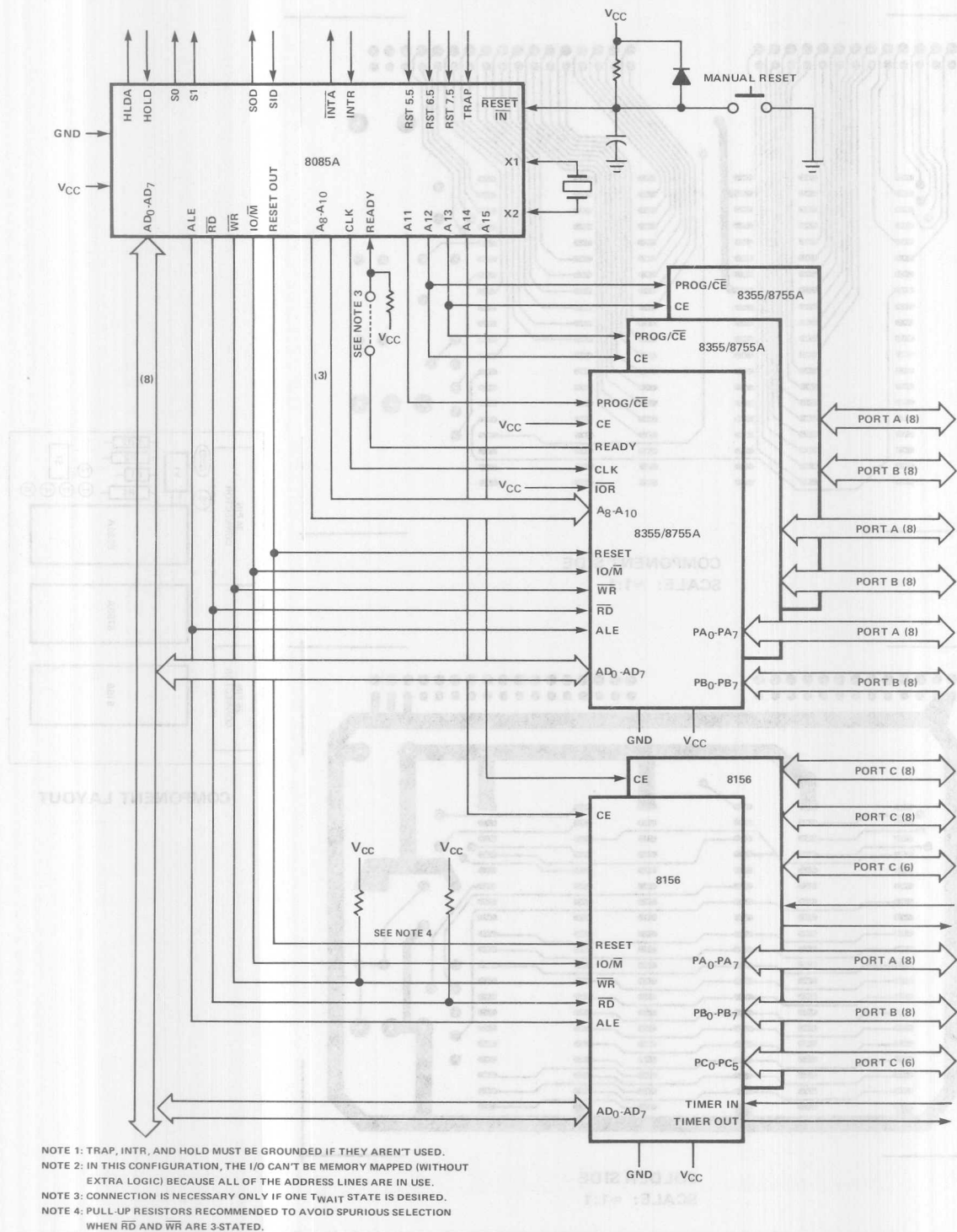


FIGURE 3-7 PRINTED CIRCUIT LAYOUT OF FIGURE 2-14A

FIGURE 3-8 EXPANDED SYSTEM

## SYSTEM OPERATION



**FIGURE 3-8 EXPANDED SYSTEM**

## 3.9 EXPANDED MCS-85™ SYSTEM

Figure 3.8 shows the circuit Figure 3.6 expanded to its maximum size without the use of any extra logic. In an extremely small board area we can fit:

PARTS	FUNCTION
1 8085A	1 CPU (Clock cycle $\leq 320$ ns)
3 8355/8755A	6144 Bytes ROM/EPROM
2 8156	512 Bytes RAM
1 Crystal	76 I/O Lines
4 Resistors	5 Interrupts
1 Capacitor	2 Programmable Timer/Counters
1 Diode	2 Serial I/O Lines
	1 Crystal and Oscillator
	1 Clock
	1 Power-on Reset

## 3.10 MCS-85 SYSTEM WITH 8185

The 8185 1K-byte static RAM chip is another multiplexed-bus component that insures that the most highly integrated systems can be built with MCS-85 components. Figure 3.9 shows a 4-chip MCS-85 system schematic with the following characteristics:

PARTS	FUNCTION
1 8085A	1 CPU
1 8185	2048 Bytes ROM/EPROM
1 8156	1280 Bytes RAM
1 8355/8755A	38 I/O Lines
	5 Interrupts
	1 Timer/Counter
	2 Serial I/O Lines

The 8185 also has power-down capability. By connecting  $\overline{CE}_1$  to  $IO/\overline{M}$  from the 8085A the 8185 will be powered down during I/O operations and Interrupt Acknowledge cycles.

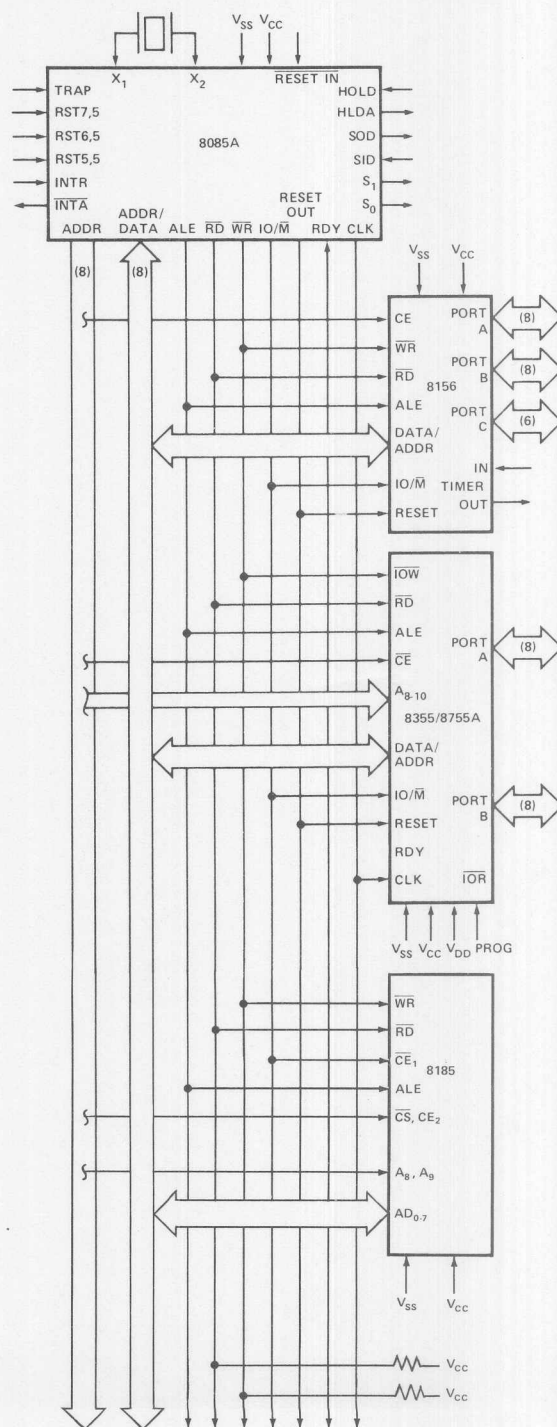


FIGURE 3-9 MCS-85 SYSTEM WITH 8185



## CHAPTER 4

### The Instruction Set

**MCS  
85**

**MCS  
85**

**MCS  
85**



## CHAPTER 4 THE INSTRUCTION SET

### 4.1 WHAT THE INSTRUCTION SET IS

A computer, no matter how sophisticated, can do only what it is instructed to do. A program is a sequence of instructions, each of which is recognized by the computer and causes it to perform an operation. Once a program is placed in memory space that is accessible to your CPU, you may run that same sequence of instructions as often as you wish to solve the same problem or to do the same function. The set of instructions to which the 8085A CPU will respond is permanently fixed in the design of the chip.

Each computer instruction allows you to initiate the performance of a specific operation. The 8085A implements a group of instructions that move data between registers, between a register and memory, and between a register and an I/O port. It also has arithmetic and logic instructions, conditional and unconditional branch instructions, and machine control instructions. The CPU recognizes these instructions only when they are coded in binary form.

### 4.2 SYMBOLS AND ABBREVIATIONS:

The following symbols and abbreviations are used in the subsequent description of the 8085A instructions:

<b>SYMBOLS</b>	<b>MEANING</b>
accumulator	Register A
addr	16-bit address quantity
data	8-bit quantity
data 16	16-bit data quantity
byte 2	The second byte of the instruction
byte 3	The third byte of the instruction
port	8-bit address of an I/O device
r,r1,r2	One of the registers A,B,C,D,E,H,L

**DDD,SSS** The bit pattern designating one of the registers A,B,C,D,E,H,L (DDD = destination, SSS = source):

<b>DDD or SSS</b>	<b>REGISTER NAME</b>
111	A
000	B
001	C
010	D
011	E
100	H
101	L

**rp** One of the register pairs:  
 B represents the B,C pair with B as the high-order register and C as the low-order register;  
 D represents the D,E pair with D as the high-order register and E as the low-order register;  
 H represents the H,L pair with H as the high-order register and L as the low-order register;  
 SP represents the 16-bit stack pointer register.

**RP** The bit pattern designating one of the register pairs B,D,H,SP:

<b>RP</b>	<b>REGISTER PAIR</b>
00	B-C
01	D-E
10	H-L
11	SP

**rh** The first (high-order) register of a designated register pair.  
**rl** The second (low-order) register of a designated register pair.

PC	16-bit program counter register (PCH and PCL are used to refer to the high-order and low-order 8 bits respectively).
SP	16-bit stack pointer register (SPH and SPL are used to refer to the high-order and low-order 8 bits respectively).
r <sub>m</sub>	Bit m of the register r (bits are number 7 through 0 from left to right).
LABEL	16-bit address of subroutine.
	The condition flags:
Z	Zero
S	Sign
P	Parity
CY	Carry
AC	Auxiliary Carry
( )	The contents of the memory location or registers enclosed in the parentheses.
←	"Is transferred to"
∧	Logical AND
⊕	Exclusive OR
∨	Inclusive OR
+	Addition
−	Two's complement subtraction
*	Multiplication
↔	"Is exchanged with"
<u>    </u>	The ones complement (e.g., $\overline{A}$ )
n	The restart number 0 through 7
NNN	The binary representation 000 through 111 for restart number 0 through 7 respectively.

The instruction set encyclopedia is a detailed description of the 8085A instruction set. Each instruction is described in the following manner:

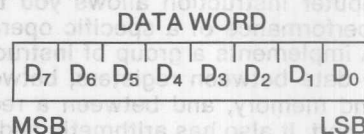
1. The MCS-85 macro assembler format, consisting of the instruction mnemonic and operand fields, is printed in **BOLDFACE** on the first line.
2. The name of the instruction is enclosed in parentheses following the mnemonic.
3. The next lines contain a symbolic description of what the instruction does.
4. This is followed by a narrative description of the operation of the instruction.

5. The boxes describe the binary codes that comprise the machine instruction.
6. The last four lines contain information about the execution of the instruction. The number of machine cycles and states required to execute the instruction are listed first. If the instruction has two possible execution times, as in a conditional jump, both times are listed, separated by a slash. Next, data addressing modes are listed if applicable. The last line lists any of the five flags that are affected by the execution of the instruction.

## 4.3 INSTRUCTION AND DATA FORMATS

Memory used in the MCS-85 system is organized in 8-bit bytes. Each byte has a unique location in physical memory. That location is described by one of a sequence of 16-bit binary addresses. The 8085A can address up to 64K ( $K = 1024$ , or  $2^{10}$ ; hence, 64K represents the decimal number 65,536) bytes of memory, which may consist of both random-access, read-write memory (RAM) and read-only memory (ROM), which is also random-access.

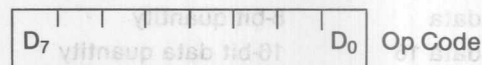
Data in the 8085A is stored in the form of 8-bit binary integers:



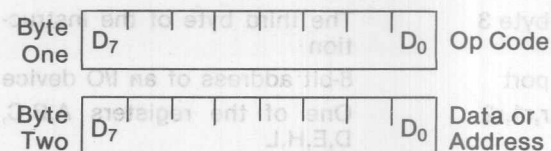
When a register or data word contains a binary number, it is necessary to establish the order in which the bits of the number are written. In the Intel 8085A, BIT 0 is referred to as the **Least Significant Bit (LSB)**, and BIT 7 (of an 8-bit number) is referred to as the **Most Significant Bit (MSB)**.

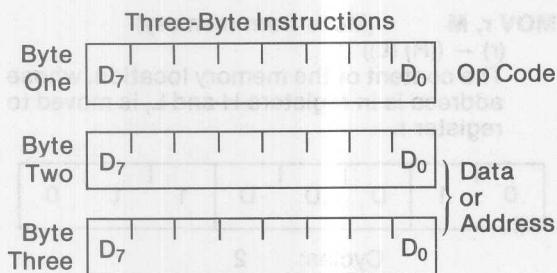
An 8085A program instruction may be one, two or three bytes in length. Multiple-byte instructions must be stored in successive memory locations; the address of the first byte is always used as the address of the instruction. The exact instruction format will depend on the particular operation to be executed.

### Single Byte Instructions



### Two-Byte Instructions





## 4.4 ADDRESSING MODES:

Often the data that is to be operated on is stored in memory. When multi-byte numeric data is used, the data, like instructions, is stored in successive memory locations, with the least significant byte first, followed by increasingly significant bytes. The 8085A has four different modes for addressing data stored in memory or in registers:

- **Direct** — Bytes 2 and 3 of the instruction contain the exact memory address of the data item (the low-order bits of the address are in byte 2, the high-order bits in byte 3).
- **Register** — The instruction specifies the register or register pair in which the data is located.
- **Register Indirect** — The instruction specifies a register pair which contains the memory address where the data is located (the high-order bits of the address are in the first register of the pair the low-order bits in the second).
- **Immediate** — The instruction contains the data itself. This is either an 8-bit quantity or a 16-bit quantity (least significant byte first, most significant byte second).

Unless directed by an interrupt or branch institution, the execution of instructions proceeds through consecutively increasing memory locations. A branch instruction can specify the address of the next instruction to be executed in one of two ways:

- **Direct** — The branch instruction contains the address of the next instruction to be executed. (Except for the 'RST' instruction, byte 2 contains the low-order address and byte 3 the high-order address.)

- **Register Indirect** — The branch instruction indicates a register-pair which contains the address of the next instruction to be executed. (The high-order bits of the address are in the first register of the pair, the low-order bits in the second.)

The RST instruction is a special one-byte call instruction (usually used during interrupt sequences). RST includes a three-bit field; program control is transferred to the instruction whose address is eight times the contents of this three-bit field.

## 4.5 CONDITION FLAGS:

There are five condition flags associated with the execution of instructions on the 8085A. They are Zero, Sign, Parity, Carry, and Auxiliary Carry. Each is represented by a 1-bit register (or flip-flop) in the CPU. A flag is set by forcing the bit to 1; it is reset by forcing the bit to 0.

Unless indicated otherwise, when an instruction affects a flag, it affects it in the following manner:

- Zero:** If the result of an instruction has the value 0, this flag is set; otherwise it is reset.
- Sign:** If the most significant bit of the result of the operation has the value 1, this flag is set; otherwise it is reset.
- Parity:** If the modulo 2 sum of the bits of the result of the operation is 0, (i.e., if the result has even parity), this flag is set; otherwise it is reset (i.e., if the result has odd parity).
- Carry:** If the instruction resulted in a carry (from addition), or a borrow (from subtraction or a comparison) out of the high-order bit, this flag is set; otherwise it is reset.

**Auxiliary Carry:** If the instruction caused a carry out of bit 3 and into bit 4 of the resulting value, the auxiliary carry is set; otherwise it is reset. This flag is affected by single-precision additions, subtractions, increments, decrements, comparisons, and logical operations, but is principally used with additions and increments preceding a DAA (Decimal Adjust Accumulator) instruction.

## 4.6 INSTRUCTION SET ENCYCLOPEDIA

In the ensuing dozen pages, the complete 8085A instruction set is described, grouped in order under five different functional headings, as follows:

- 1. Data Transfer Group** — Moves data between registers or between memory locations and registers. Includes moves, loads, stores, and exchanges. (See below.)
- 2. Arithmetic Group** — Adds, subtracts, increments, or decrements data in registers or memory. (See page 4-13.)
- 3. Logic Group** — ANDs, ORs, XORs, compares, rotates, or complements data in registers or between memory and a register. (See page 4-16.)
- 4. Branch Group** — Initiates conditional or unconditional jumps, calls, returns, and restarts. (See page 4-20.)
- 5. Stack, I/O, and Machine Control Group** — Includes instructions for maintaining the stack, reading from input ports, writing to output ports, setting and reading interrupt masks, and setting and clearing flags. (See page 4-22.)

The formats described in the encyclopedia reflect the assembly language processed by Intel-supplied assembler, used with the Intel<sup>®</sup> development systems.

### 4.6.1 Data Transfer Group

This group of instructions transfers data to and from registers and memory. **Condition flags are not affected by any instruction in this group.**

#### MOV r1, r2 (Move Register)

(r1) ← (r2)

The content of register r2 is moved to register r1.

0	1	D	D	D	S	S	S
---	---	---	---	---	---	---	---

Cycles: 1  
States: 4  
Addressing: register  
Flags: none

#### MOV r, M (Move from memory)

(r) ← ((H) (L))

The content of the memory location, whose address is in registers H and L, is moved to register r.

0	1	D	D	D	1	1	0
---	---	---	---	---	---	---	---

Cycles: 2  
States: 7  
Addressing: reg. indirect  
Flags: none

#### MOV M, r (Move to memory)

((H) (L)) ← (r)

The content of register r is moved to the memory location whose address is in registers H and L.

0	1	1	1	0	S	S	S
---	---	---	---	---	---	---	---

Cycles: 2  
States: 7  
Addressing: reg. indirect  
Flags: none

#### MVI r, data (Move Immediate)

(r) ← (byte 2)

The content of byte 2 of the instruction is moved to register r.

0	0	D	D	D	1	1	0
data							

Cycles: 2  
States: 7  
Addressing: immediate  
Flags: none

#### MVI M, data (Move to memory immediate)

((H) (L)) ← (byte 2)

The content of byte 2 of the instruction is moved to the memory location whose address is in registers H and L.

0	0	1	1	0	1	1	0
data							

Cycles: 3  
States: 10  
Addressing: immed./reg. indirect  
Flags: none

## THE INSTRUCTION SET

**LXI rp, data 16** (Load register pair immediate)  
 (rh) ← (byte 3),  
 (rl) ← (byte 2)  
 Byte 3 of the instruction is moved into the high-order register (rh) of the register pair rp. Byte 2 of the instruction is moved into the low-order register (rl) of the register pair rp.

0	0	R	P	0	0	0	1
low-order data							
high-order data							

Cycles: 3  
 States: 10  
 Addressing: immediate  
 Flags: none

**LDA addr** (Load Accumulator direct)  
 (A) ← ((byte 3)(byte 2))  
 The content of the memory location, whose address is specified in byte 2 and byte 3 of the instruction, is moved to register A.

0	0	1	1	1	0	1	0
low-order addr							
high-order addr							

Cycles: 4  
 States: 13  
 Addressing: direct  
 Flags: none

**STA addr** (Store Accumulator direct)  
 ((byte 3)(byte 2)) ← (A)  
 The content of the accumulator is moved to the memory location whose address is specified in byte 2 and byte 3 of the instruction.

0	0	1	1	0	0	1	0
low-order addr							
high-order addr							

Cycles: 4  
 States: 13  
 Addressing: direct  
 Flags: none

**LHLD addr** (Load H and L direct)  
 (L) ← ((byte 3)(byte 2))  
 (H) ← ((byte 3)(byte 2) + 1)  
 The content of the memory location, whose address is specified in byte 2 and byte 3 of the instruction, is moved to register L. The content of the memory location at the succeeding address is moved to register H.

0	0	1	0	1	0	1	0
low-order addr							
high-order addr							

Cycles: 5  
 States: 16  
 Addressing: direct  
 Flags: none

**SHLD addr** (Store H and L direct)  
 ((byte 3)(byte 2)) ← (L)  
 ((byte 3)(byte 2) + 1) ← (H)  
 The content of register L is moved to the memory location whose address is specified in byte 2 and byte 3. The content of register H is moved to the succeeding memory location.

0	0	1	0	0	0	1	0
low-order addr							
high-order addr							

Cycles: 5  
 States: 16  
 Addressing: direct  
 Flags: none

**LDAX rp** (Load accumulator indirect)  
 (A) ← ((rp))  
 The content of the memory location, whose address is in the register pair rp, is moved to register A. Note: only register pairs rp=B (registers B and C) or rp=D (registers D and E) may be specified.

0	0	R	P	1	0	1	0
---	---	---	---	---	---	---	---

Cycles: 2  
 States: 7  
 Addressing: reg. indirect  
 Flags: none

## THE INSTRUCTION SET

memory location whose address is in the register pair *rp*. Note: only register pairs *rp*=B (registers B and C) or *rp*=D (registers D and E) may be specified.

0	0	R	P	0	0	1	0
---	---	---	---	---	---	---	---

Cycles: 2  
States: 7  
Addressing: reg. indirect  
Flags: none

### XCHG (Exchange H and L with D and E)

(H) ↔ (D)

(L) ↔ (E)

The contents of registers H and L are exchanged with the contents of registers D and E.

1	1	1	0	1	0	1	1
---	---	---	---	---	---	---	---

Cycles: 1  
States: 4  
Addressing: register  
Flags: none

### 4.6.2 Arithmetic Group

This group of instructions performs arithmetic operations on data in registers and memory.

**Unless indicated otherwise, all instructions in this group affect the Zero, Sign, Parity, Carry, and Auxiliary Carry flags according to the standard rules.**

All subtraction operations are performed via two's complement arithmetic and set the carry flag to one to indicate a borrow and clear it to indicate no borrow.

### ADD r (Add Register)

(A) ← (A) + (r)

The content of register *r* is added to the content of the accumulator. The result is placed in the accumulator.

1	0	0	0	0	S	S	S
---	---	---	---	---	---	---	---

Cycles: 1  
States: 4  
Addressing: register  
Flags: Z,S,P,CY,AC

address is contained in the H and L registers is added to the content of the accumulator. The result is placed in the accumulator.

1	0	0	0	0	1	1	0
---	---	---	---	---	---	---	---

Cycles: 2  
States: 7  
Addressing: reg. indirect  
Flags: Z,S,P,CY,AC

### ADI data (Add immediate)

(A) ← (A) + (byte 2)

The content of the second byte of the instruction is added to the content of the accumulator. The result is placed in the accumulator.

1	1	0	0	0	1	1	0
---	---	---	---	---	---	---	---

Cycles: 2  
States: 7  
Addressing: immediate  
Flags: Z,S,P,CY,AC

### ADC r (Add Register with carry)

(A) ← (A) + (r) + (CY)

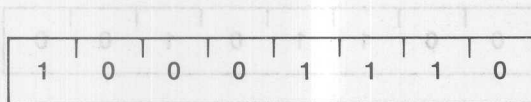
The content of register *r* and the content of the carry bit are added to the content of the accumulator. The result is placed in the accumulator.

1	0	0	0	1	S	S	S
---	---	---	---	---	---	---	---

Cycles: 1  
States: 4  
Addressing: register  
Flags: Z,S,P,CY,AC

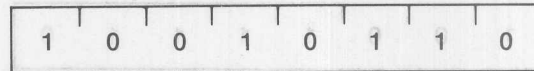
## THE INSTRUCTION SET

**ADC M** (Add memory with carry)  
 $(A) \leftarrow (A) + ((H) (L)) + (CY)$   
 The content of the memory location whose address is contained in the H and L registers and the content of the CY flag are added to the accumulator. The result is placed in the accumulator.



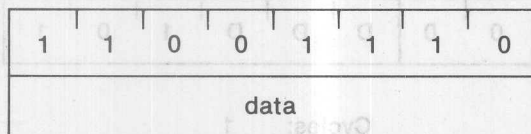
Cycles: 2  
 States: 7  
 Addressing: reg. indirect  
 Flags: Z,S,P,CY,AC

**SUB M** (Subtract memory)  
 $(A) \leftarrow (A) - ((H) (L))$   
 The content of the memory location whose address is contained in the H and L registers is subtracted from the content of the accumulator. The result is placed in the accumulator.



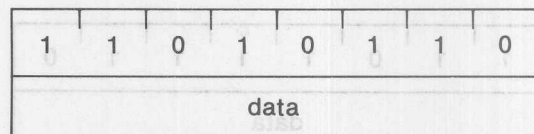
Cycles: 2  
 States: 7  
 Addressing: reg. indirect  
 Flags: Z,S,P,CY,AC

**ACI data** (Add immediate with carry)  
 $(A) \leftarrow (A) + (\text{byte 2}) + (CY)$   
 The content of the second byte of the instruction and the content of the CY flag are added to the contents of the accumulator. The result is placed in the accumulator.



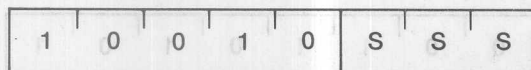
Cycles: 2  
 States: 7  
 Addressing: immediate  
 Flags: Z,S,P,CY,AC

**SUI data** (Subtract immediate)  
 $(A) \leftarrow (A) - (\text{byte 2})$   
 The content of the second byte of the instruction is subtracted from the content of the accumulator. The result is placed in the accumulator.



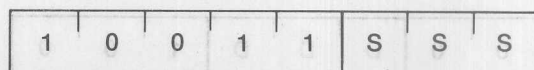
Cycles: 2  
 States: 7  
 Addressing: immediate  
 Flags: Z,S,P,CY,AC

**SUB r** (Subtract Register)  
 $(A) \leftarrow (A) - (r)$   
 The content of register r is subtracted from the content of the accumulator. The result is placed in the accumulator.



Cycles: 1  
 States: 4  
 Addressing: register  
 Flags: Z,S,P,CY,AC

**SBB r** (Subtract Register with borrow)  
 $(A) \leftarrow (A) - (r) - (CY)$   
 The content of register r and the content of the CY flag are both subtracted from the accumulator. The result is placed in the accumulator.



Cycles: 1  
 States: 4  
 Addressing: register  
 Flags: Z,S,P,CY,AC

## THE INSTRUCTION SET

### SBB M (Subtract memory with borrow)

$$(A) \leftarrow (A) - ((H) (L)) - (CY)$$

The content of the memory location whose address is contained in the H and L registers and the content of the CY flag are both subtracted from the accumulator. The result is placed in the accumulator.

1	0	0	1	1	1	1	1	0
---	---	---	---	---	---	---	---	---

Cycles: 2  
States: 7  
Addressing: reg. indirect  
Flags: Z,S,P,CY,AC

### INR M (Increment memory)

$$((H) (L)) \leftarrow ((H) (L)) + 1$$

The content of the memory location whose address is contained in the H and L registers is incremented by one. Note: All condition flags **except** CY are affected.

0	0	1	1	0	1	0	0
---	---	---	---	---	---	---	---

Cycles: 3  
States: 10  
Addressing: reg. indirect  
Flags: Z,S,P,AC

### SBI data (Subtract immediate with borrow)

$$(A) \leftarrow (A) - (\text{byte 2}) - (CY)$$

The contents of the second byte of the instruction and the contents of the CY flag are both subtracted from the accumulator. The result is placed in the accumulator.

1	1	0	1	1	1	1	1	0
data								

Cycles: 2  
States: 7  
Addressing: immediate  
Flags: Z,S,P,CY,AC

### DCR r (Decrement Register)

$$(r) \leftarrow (r) - 1$$

The content of register r is decremented by one. Note: All condition flags **except** CY are affected.

0	0	D	D	D	1	0	1
---	---	---	---	---	---	---	---

Cycles: 1  
States: 4  
Addressing: register  
Flags: Z,S,P,AC

### INR r (Increment Register)

$$(r) \leftarrow (r) + 1$$

The content of register r is incremented by one. Note: All condition flags **except** CY are affected.

0	0	D	D	D	1	0	0
---	---	---	---	---	---	---	---

Cycles: 1  
States: 4  
Addressing: register  
Flags: Z,S,P,AC

### DCR M (Decrement memory)

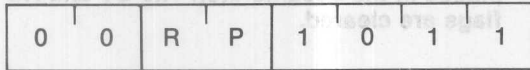
$$((H) (L)) \leftarrow ((H) (L)) - 1$$

The content of the memory location whose address is contained in the H and L registers is decremented by one. Note: All condition flags **except** CY are affected.

0	0	1	1	0	1	0	1
---	---	---	---	---	---	---	---

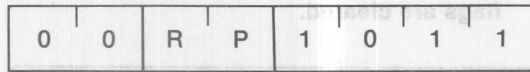
Cycles: 3  
States: 10  
Addressing: reg. indirect  
Flags: Z,S,P,AC

**INX rp** (Increment register pair)  
 $(rh)(rl) \leftarrow (rh)(rl) + 1$   
 The content of the register pair rp is incremented by one. Note: **No condition flags are affected.**



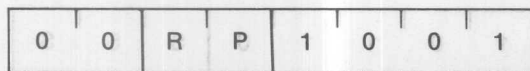
Cycles: 1  
 States: 6  
 Addressing: register  
 Flags: none

**DCX rp** (Decrement register pair)  
 $(rh)(rl) \leftarrow (rh)(rl) - 1$   
 The content of the register pair rp is decremented by one. Note: **No condition flags are affected.**



Cycles: 1  
 States: 6  
 Addressing: register  
 Flags: none

**DAD rp** (Add register pair to H and L)  
 $(H)(L) \leftarrow (H)(L) + (rh)(rl)$   
 The content of the register pair rp is added to the content of the register pair H and L. The result is placed in the register pair H and L. Note: **Only the CY flag is affected.** It is set if there is a carry out of the double precision add; otherwise it is reset.

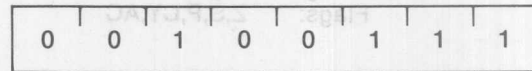


Cycles: 3  
 States: 10  
 Addressing: register  
 Flags: CY

**DAA** (Decimal Adjust Accumulator)  
 The eight-bit number in the accumulator is adjusted to form two four-bit Binary-Coded-Decimal digits by the following process:

1. If the value of the least significant 4 bits of the accumulator is greater than 9 or if the AC flag is set, 6 is added to the accumulator.
2. If the value of the most significant 4 bits of the accumulator is now greater than 9, or if the CY flag is set, 6 is added to the most significant 4 bits of the accumulator.

NOTE: All flags are affected.



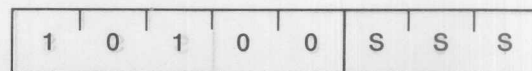
Cycles: 1  
 States: 4  
 Flags: Z,S,P,CY,AC

#### 4.6.3 Logic Group

This group of instructions performs logical (Boolean) operations on data in registers and memory and on condition flags.

Unless indicated otherwise, all instructions in this group affect the Zero, Sign, Parity, Auxiliary Carry, and Carry flags according to the standard rules.

**ANA r** (AND Register)  
 $(A) \leftarrow (A) \wedge (r)$   
 The content of register r is logically ANDed with the content of the accumulator. The result is placed in the accumulator. **The CY flag is cleared and AC is set.**



Cycles: 1  
 States: 4  
 Addressing: register  
 Flags: Z,S,P,CY,AC

## THE INSTRUCTION SET

### ANA M (AND memory)

$$(A) \leftarrow (A) \wedge ((H) (L))$$

The contents of the memory location whose address is contained in the H and L registers is logically ANDed with the content of the accumulator. The result is placed in the accumulator. **The CY flag is cleared and AC is set.**

1	0	1	0	0	1	1	0
---	---	---	---	---	---	---	---

Cycles: 2  
 States: 7  
 Addressing: reg. indirect  
 Flags: Z,S,P,CY,AC

### XRA M (Exclusive OR Memory)

$$(A) \leftarrow (A) \vee ((H) (L))$$

The content of the memory location whose address is contained in the H and L registers is exclusive-OR'd with the content of the accumulator. The result is placed in the accumulator. **The CY and AC flags are cleared.**

1	0	1	0	1	1	1	0
---	---	---	---	---	---	---	---

Cycles: 2  
 States: 7  
 Addressing: reg. indirect  
 Flags: Z,S,P,CY,AC

### ANI data (AND immediate)

$$(A) \leftarrow (A) \wedge (\text{byte } 2)$$

The content of the second byte of the instruction is logically ANDed with the contents of the accumulator. The result is placed in the accumulator. **The CY flag is cleared and AC is set.**

1	1	1	0	0	1	1	0
data							

Cycles: 2  
 States: 7  
 Addressing: immediate  
 Flags: Z,S,P,CY,AC

### XRI data (Exclusive OR immediate)

$$(A) \leftarrow (A) \vee (\text{byte } 2)$$

The content of the second byte of the instruction is exclusive-OR'd with the content of the accumulator. The result is placed in the accumulator. **The CY and AC flags are cleared.**

1	1	1	0	1	1	1	0
data							

Cycles: 2  
 States: 7  
 Addressing: immediate  
 Flags: Z,S,P,CY,AC

### XRA r (Exclusive OR Register)

$$(A) \leftarrow (A) \vee (r)$$

The content of register r is exclusive-OR'd with the content of the accumulator. The result is placed in the accumulator. **The CY and AC flags are cleared.**

1	0	1	0	1	S	S	S
---	---	---	---	---	---	---	---

Cycles: 1  
 States: 4  
 Addressing: register  
 Flags: Z,S,P,CY,AC

### ORA r (OR Register)

$$(A) \leftarrow (A) \vee (r)$$

The content of register r is inclusive-OR'd with the content of the accumulator. The result is placed in the accumulator. **The CY and AC flags are cleared.**

1	0	1	1	0	S	S	S
---	---	---	---	---	---	---	---

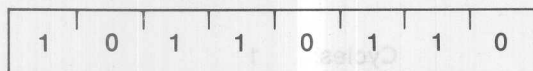
Cycles: 1  
 States: 4  
 Addressing: register  
 Flags: Z,S,P,CY,AC

## THE INSTRUCTION SET

### ORA M (OR memory)

(A) ← (A) V ((H) (L))

The content of the memory location whose address is contained in the H and L registers is inclusive-OR'd with the content of the accumulator. The result is placed in the accumulator. **The CY and AC flags are cleared.**

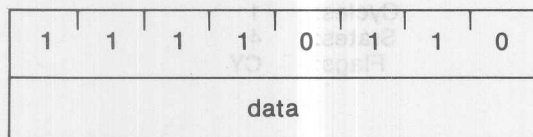


Cycles: 2  
 States: 7  
 Addressing: reg. indirect  
 Flags: Z,S,P,CY,AC

### ORI data (OR Immediate)

(A) ← (A) V (byte 2)

The content of the second byte of the instruction is inclusive-OR'd with the content of the accumulator. The result is placed in the accumulator. **The CY and AC flags are cleared.**

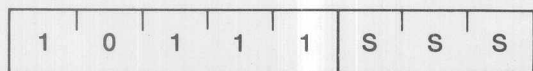


Cycles: 2  
 States: 7  
 Addressing: immediate  
 Flags: Z,S,P,CY,AC

### CMP r (Compare Register)

(A) ← (r)

The content of register r is subtracted from the accumulator. The accumulator remains unchanged. The condition flags are set as a result of the subtraction. **The Z flag is set to 1 if (A) = (r). The CY flag is set to 1 if (A) < (r).**

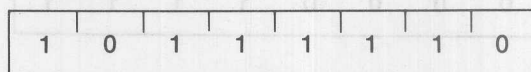


Cycles: 1  
 States: 4  
 Addressing: register  
 Flags: Z,S,P,CY,AC

### CMP M (Compare memory)

(A) ← ((H) (L))

The content of the memory location whose address is contained in the H and L registers is subtracted from the accumulator. The accumulator remains unchanged. The condition flags are set as a result of the subtraction. **The Z flag is set to 1 if (A) = ((H) (L)). The CY flag is set to 1 if (A) < ((H) (L)).**

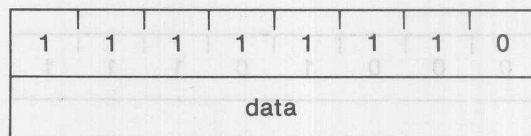


Cycles: 2  
 States: 7  
 Addressing: reg. indirect  
 Flags: Z,S,P,CY,AC

### CPI data (Compare immediate)

(A) ← (byte 2)

The content of the second byte of the instruction is subtracted from the accumulator. The condition flags are set by the result of the subtraction. **The Z flag is set to 1 if (A) = (byte 2). The CY flag is set to 1 if (A) < (byte 2).**



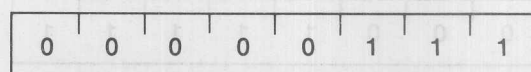
Cycles: 2  
 States: 7  
 Addressing: immediate  
 Flags: Z,S,P,CY,AC

### RLC (Rotate left)

(A<sub>n+1</sub>) ← (A<sub>n</sub>) ; (A<sub>0</sub>) ← (A<sub>7</sub>)

(CY) ← (A<sub>7</sub>)

The content of the accumulator is rotated left one position. The low order bit and the CY flag are both set to the value shifted out of the high order bit position. **Only the CY flag is affected.**



Cycles: 1  
 States: 4  
 Flags: CY

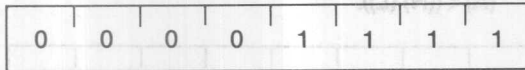
## THE INSTRUCTION SET

### RRC (Rotate right)

$(A_n) \leftarrow (A_{n+1}); (A_7) \leftarrow (A_0)$

$(CY) \leftarrow (A_0)$

The content of the accumulator is rotated right one position. The high order bit and the CY flag are both set to the value shifted out of the low order bit position. **Only the CY flag is affected.**



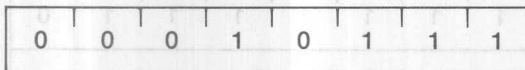
Cycles: 1  
States: 4  
Flags: CY

### RAL (Rotate left through carry)

$(A_{n+1}) \leftarrow (A_n); (CY) \leftarrow (A_7)$

$(A_0) \leftarrow (CY)$

The content of the accumulator is rotated left one position through the CY flag. The low order bit is set equal to the CY flag and the CY flag is set to the value shifted out of the high order bit. **Only the CY flag is affected.**



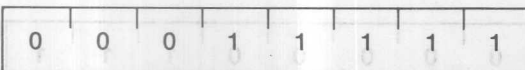
Cycles: 1  
States: 4  
Flags: CY

### RAR (Rotate right through carry)

$(A_n) \leftarrow (A_{n+1}); (CY) \leftarrow (A_0)$

$(A_7) \leftarrow (CY)$

The content of the accumulator is rotated right one position through the CY flag. The high order bit is set to the CY flag and the CY flag is set to the value shifted out of the low order bit. **Only the CY flag is affected.**

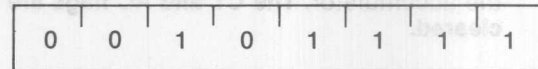


Cycles: 1  
States: 4  
Flags: CY

### CMA (Complement accumulator)

$(A) \leftarrow (\bar{A})$

The contents of the accumulator are complemented (zero bits become 1, one bits become 0). **No flags are affected.**

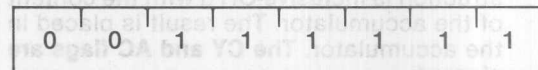


Cycles: 1  
States: 4  
Flags: none

### CMC (Complement carry)

$(CY) \leftarrow (\bar{CY})$

The CY flag is complemented. **No other flags are affected.**

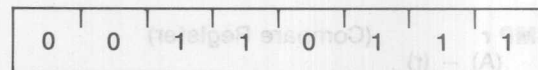


Cycles: 1  
States: 4  
Flags: CY

### STC (Set carry)

$(CY) \leftarrow 1$

The CY flag is set to 1. **No other flags are affected.**



Cycles: 1  
States: 4  
Flags: CY

## 4.6.4 Branch Group

This group of instructions alter normal sequential program flow.

**Condition flags are not affected** by any instruction in this group.

The two types of branch instructions are unconditional and conditional. Unconditional transfers simply perform the specified operation on register PC (the program counter). Conditional transfers examine the status of one of the four processor flags to determine if the specified branch is to be executed. The conditions that may be specified are as follows:

CONDITION	CCC
NZ — not zero (Z = 0)	000
Z — zero (Z = 1)	001
NC — no carry (CY = 0)	010
C — carry (CY = 1)	011
PO — parity odd (P = 0)	100
PE — parity even (P = 1)	101
P — plus (S = 0)	110
M — minus (S = 1)	111

### JMP addr (Jump)

(PC) ← (byte 3) (byte 2)

Control is transferred to the instruction whose address is specified in byte 3 and byte 2 of the current instruction.

1	1	0	0	0	0	1	1
low-order addr							
high-order addr							

Cycles: 3  
States: 10  
Addressing: immediate  
Flags: none

### Jcondition addr (Conditional jump)

If (CCC),

(PC) ← (byte 3) (byte 2)

If the specified condition is true, control is transferred to the instruction whose address is specified in byte 3 and byte 2 of the current instruction; otherwise, control continues sequentially.

1	1	C	C	C	0	1	0
low-order addr							
high-order addr							

Cycles: 2/3  
States: 7/10  
Addressing: immediate  
Flags: none

### CALL addr (Call)

((SP) - 1) ← (PCH)

((SP) - 2) ← (PCL)

(SP) ← (SP) - 2

(PC) ← (byte 3) (byte 2)

The high-order eight bits of the next instruction address are moved to the memory location whose address is one less than the content of register SP. The low-order eight bits of the next instruction address are moved to the memory location whose address is two less than the content of register SP. The content of register SP is decremented by 2. Control is transferred to the instruction whose address is specified in byte 3 and byte 2 of the current instruction.

1	1	0	0	1	1	0	1
low-order addr							
high-order addr							

Cycles: 5  
States: 18  
Addressing: immediate/  
reg. indirect  
Flags: none

$((SP) - 2) - (PCL)$

$(SP) - (SP) - 2$

$(PC) - (\text{byte 3}) (\text{byte 2})$

If the specified condition is true, the actions specified in the CALL instruction (see above) are performed; otherwise, control continues sequentially.

1	1	C	C	C	1	0	0
low-order addr							
high-order addr							

Cycles: 2/5

States: 9/18

Addressing: immediate/  
reg. indirect

Flags: none

**RET** (Return)

$(PCL) - ((SP));$

$(PCH) - ((SP) + 1);$

$(SP) - (SP) + 2;$

The content of the memory location whose address is specified in register SP is moved to the low-order eight bits of register PC. The content of the memory location whose address is one more than the content of register SP is moved to the high-order eight bits of register PC. The content of register SP is incremented by 2.

1	1	0	0	1	0	0	1
---	---	---	---	---	---	---	---

Cycles: 3

States: 10

Addressing: reg. indirect

Flags: none

$(PCH) - ((SP))$

$(PCH) - ((SP) + 1)$

$(SP) - (SP) + 2$

If the specified condition is true, the actions specified in the RET instruction (see above) are performed; otherwise, control continues sequentially.

1	1	C	C	C	0	0	0
---	---	---	---	---	---	---	---

Cycles: 1/3

States: 6/12

Addressing: reg. indirect

Flags: none

**RST n** (Restart)

$((SP) - 1) - (PCH)$

$((SP) - 2) - (PCL)$

$(SP) - (SP) - 2$

$(PC) - 8 * (NNN)$

The high-order eight bits of the next instruction address are moved to the memory location whose address is one less than the content of register SP. The low-order eight bits of the next instruction address are moved to the memory location whose address is two less than the content of register SP. The content of register SP is decremented by two. Control is transferred to the instruction whose address is eight times the content of NNN.

1	1	N	N	N	1	1	1
---	---	---	---	---	---	---	---

Cycles: 3

States: 12

Addressing: reg. indirect

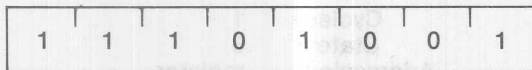
Flags: none

15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0

0	0	0	0	0	0	0	0	0	0	N	N	N	0	0	0
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

Program Counter After Restart

**PCHL** (Jump H and L indirect — move H and L to PC)  
 $(PCH) \leftarrow (H)$   
 $(PCL) \leftarrow (L)$   
 The content of register H is moved to the high-order eight bits of register PC. The content of register L is moved to the low-order eight bits of register PC.



Cycles: 1  
 States: 6  
 Addressing: register  
 Flags: none

## 4.6.5 Stack, I/O, and Machine Control Group

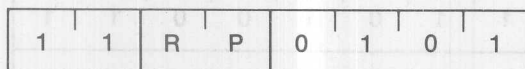
This group of instructions performs I/O, manipulates the Stack, and alters internal control flags.

Unless otherwise specified, **condition flags are not affected by any instructions in this group.**

**PUSH rp** (Push)

$((SP) - 1) \leftarrow (rh)$   
 $((SP) - 2) \leftarrow (rl)$   
 $((SP) - (SP) - 2$

The content of the high-order register of register pair *rp* is moved to the memory location whose address is one less than the content of register SP. The content of the low-order register of register pair *rp* is moved to the memory location whose address is two less than the content of register SP. The content of register SP is decremented by 2. **Note: Register pair *rp* = SP may not be specified.**

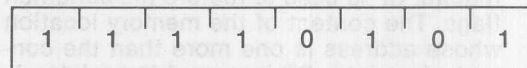


Cycles: 3  
 States: 12  
 Addressing: reg. indirect  
 Flags: none

**PUSH PSW** (Push processor status word)

$((SP) - 1) \leftarrow (A)$   
 $((SP) - 2)_0 \leftarrow (CY), ((SP) - 2)_1 \leftarrow X$   
 $((SP) - 2)_2 \leftarrow (P), ((SP) - 2)_3 \leftarrow X$   
 $((SP) - 2)_4 \leftarrow (AC), ((SP) - 2)_5 \leftarrow X$   
 $((SP) - 2)_6 \leftarrow (Z), ((SP) - 2)_7 \leftarrow (S)$   
 $(SP) \leftarrow (SP) - 2$  X: Undefined.

The content of register A is moved to the memory location whose address is one less than register SP. The contents of the condition flags are assembled into a processor status word and the word is moved to the memory location whose address is two less than the content of register SP. The content of register SP is decremented by two.



Cycles: 3  
 States: 12  
 Addressing: reg. indirect  
 Flags: none

## FLAG WORD

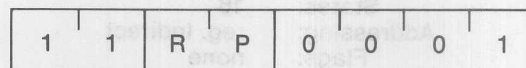
D <sub>7</sub>	D <sub>6</sub>	D <sub>5</sub>	D <sub>4</sub>	D <sub>3</sub>	D <sub>2</sub>	D <sub>1</sub>	D <sub>0</sub>
S	Z	X	AC	X	P	X	CY

X: undefined

**POP rp** (POP)

$(rl) \leftarrow ((SP))$   
 $(rh) \leftarrow ((SP) + 1)$   
 $(SP) \leftarrow (SP) + 2$

The content of the memory location, whose address is specified by the content of register SP, is moved to the low-order register of register pair *rp*. The content of the memory location, whose address is one more than the content of register SP, is moved to the high-order register of register *rp*. The content of register SP is incremented by 2. **Note: Register pair *rp* = SP may not be specified.**



Cycles: 3  
 States: 10  
 Addressing: reg. indirect  
 Flags: none

## THE INSTRUCTION SET

### POP PSW (Pop processor status word)

(CY) ← ((SP))<sub>0</sub>

(P) ← ((SP))<sub>2</sub>

(AC) ← ((SP))<sub>4</sub>

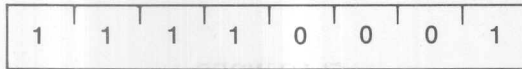
(Z) ← ((SP))<sub>6</sub>

(S) ← ((SP))<sub>7</sub>

(A) ← ((SP) + 1)

(SP) ← (SP) + 2

The content of the memory location whose address is specified by the content of register SP is used to restore the condition flags. The content of the memory location whose address is one more than the content of register SP is moved to register A. The content of register SP is incremented by 2.



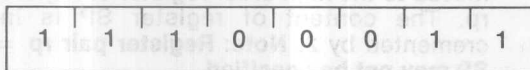
Cycles: 3  
States: 10  
Addressing: reg. indirect  
Flags: Z,S,P,CY,AC

### XTHL (Exchange stack top with H and L)

(L) ↔ ((SP))

(H) ↔ ((SP) + 1)

The content of the L register is exchanged with the content of the memory location whose address is specified by the content of register SP. The content of the H register is exchanged with the content of the memory location whose address is one more than the content of register SP.

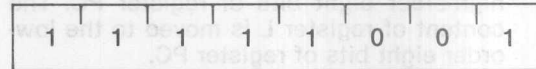


Cycles: 5  
States: 16  
Addressing: reg. indirect  
Flags: none

### SPHL (Move HL to SP)

(SP) ← (H) (L)

The contents of registers H and L (16 bits) are moved to register SP.

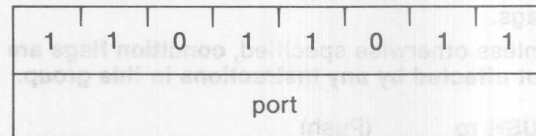


Cycles: 1  
States: 6  
Addressing: register  
Flags: none

### IN port (Input)

(A) ← (data)

The data placed on the eight bit bi-directional data bus by the specified port is moved to register A.

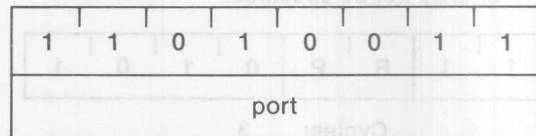


Cycles: 3  
States: 10  
Addressing: direct  
Flags: none

### OUT port (Output)

(data) ← (A)

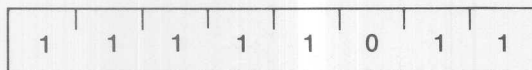
The content of register A is placed on the eight bit bi-directional data bus for transmission to the specified port.



Cycles: 3  
States: 10  
Addressing: direct  
Flags: none

## THE INSTRUCTION SET

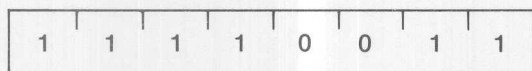
**EI** (Enable interrupts)  
The interrupt system is enabled **following the execution of the next instruction.**



Cycles: 1  
States: 4  
Flags: none

**NOTE:** Interrupts are not recognized during the EI instruction. Placing an EI instruction on the bus in response to INTA during an INA cycle is prohibited.

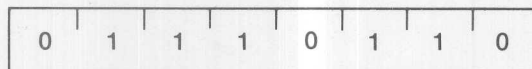
**DI** (Disable interrupts)  
The interrupt system is disabled **immediately following the execution of the DI instruction.**



Cycles: 1  
States: 4  
Flags: none

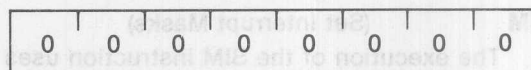
**NOTE:** Interrupts are not recognized during the DI instruction. Placing a DI instruction on the bus in response to INTA during an INA cycle is prohibited.

**HLT** (Halt)  
The processor is stopped. The registers and flags are unaffected. A second ALE is generated during the execution of HLT to strobe out the Halt cycle status information.



Cycles: 1+  
States: 5  
Flags: none

**NOP** (No op)  
No operation is performed. The registers and flags are unaffected.



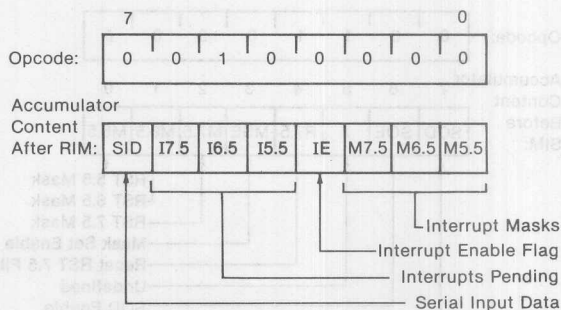
Cycles: 1  
States: 4  
Flags: none

**RIM** (Read Interrupt Masks)

The RIM instruction loads data into the accumulator relating to interrupts and the serial input. This data contains the following information:

- Current interrupt mask status for the RST 5.5, 6.5, and 7.5 hardware interrupts (1 = mask disabled)
- Current interrupt enable flag status (1 = interrupts enabled) except immediately following a TRAP interrupt. (See below.)
- Hardware interrupts pending (i.e., signal received but not yet serviced), on the RST 5.5, 6.5, and 7.5 lines.
- Serial input data.

Immediately following a TRAP interrupt, the RIM instruction must be executed as a part of the service routine if you need to retrieve current interrupt status later. Bit 3 of the accumulator is (in this special case only) loaded with the interrupt enable (IE) flag status that existed prior to the TRAP interrupt. Following an RST 5.5, 6.5, 7.5, or INTR interrupt, the interrupt flag flip-flop reflects the current interrupt enable status. Bit 6 of the accumulator (I7.5) is loaded with the status of the RST 7.5 flip-flop, which is always set (edge-triggered) by an input on the RST 7.5 input line, even when that interrupt has been previously masked. (See SIM Instruction.)



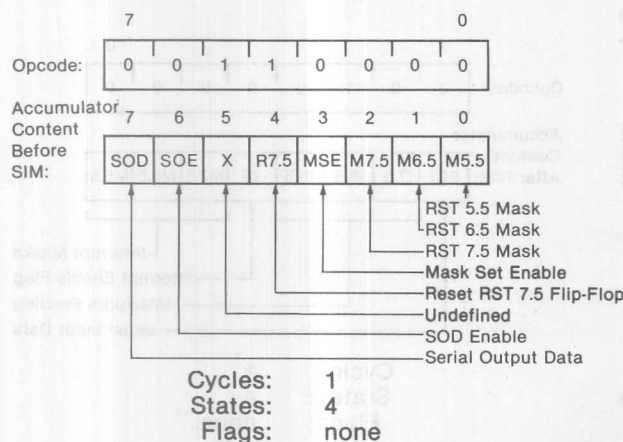
Cycles: 1  
States: 4  
Flags: none

The execution of the SIM instruction uses the contents of the accumulator (which must be previously loaded) to perform the following functions:

- Program the interrupt mask for the RST 5.5, 6.5, and 7.5 hardware interrupts.
- Reset the edge-triggered RST 7.5 input latch.
- Load the SOD output latch.

To program the interrupt masks, first set accumulator bit 3 to 1 and set to 1 any bits 0, 1, and 2, which disable interrupts RST 5.5, 6.5, and 7.5, respectively. Then do a SIM instruction. If accumulator bit 3 is 0 when the SIM instruction is executed, the interrupt mask register will not change. If accumulator bit 4 is 1 when the SIM instruction is executed, the RST 7.5 latch is then reset. RST 7.5 is distinguished by the fact that its latch is always set by a rising edge on the RST 7.5 input pin, even if the jump to service routine is inhibited by masking. This latch remains high until cleared by a RESET IN, by a SIM Instruction with accumulator bit 4 high, or by an internal processor acknowledge to an RST 7.5 interrupt subsequent to the removal of the mask (by a SIM instruction). The RESET IN signal always sets all three RST mask bits.

If accumulator bit 6 is at the 1 level when the SIM instruction is executed, the state of accumulator bit 7 is loaded into the SOD latch and thus becomes available for interface to an external device. The SOD latch is unaffected by the SIM instruction if bit 6 is 0. SOD is always reset by the RESET IN signal.



## 8085A INSTRUCTION SET INDEX

Table 4-1

Instruction	Code	Bytes	T States	Machine Cycles	Page	Instruction	Code	Bytes	T States	Machine Cycles	Page
ACI DATA	CE data	2	7	F R	4-7	LXI RP,DATA16	00RP 0001 data16	3	10	F R R	4-5
ADC REG	1000 1SSS	1	4	F	4-6	MOV REG,REG	01DD DSSS	1	4	F	4-4
ADC M	8E	1	7	F R	4-7	MOV M,REG	0111 0SSS	1	7	FW	4-4
ADD REG	1000 0SSS	1	4	F	4-6	MOV REG,M	01DD D110	1	7	F R	4-4
ADD M	86	1	7	F R	4-6	MVI REG,DATA	00DD D110 data	2	7	F R	4-4
ADI DATA	C6 data	2	7	F R	4-6	MVI M,DATA	36 data	2	10	F R W	4-4
ANA REG	1010 0SSS	1	4	F	4-9	NOP	00	1	4	F	4-17
ANA M	A6	1	7	F R	4-10	ORA REG	1011 0SSS	1	4	F	4-10
ANI DATA	E6 data	2	7	F R	4-10	ORA M	86	1	7	F R	4-11
CALL LABEL	CD addr	3	18	S R R W W	4-13	ORI DATA	F6 data	2	7	F R	4-11
CC LABEL	DC addr	3	9/18	S R/S R R W W	4-14	OUT PORT	D3 data	2	10	F R O	4-16
CM LABEL	FC addr	3	9/18	S R/S R R W W	4-14	PCHL	E9	1	6	S	4-15
CMA	2F	1	4	F	4-12	POP RP	11RP 0001	1	10	F R R	4-15
CMC	3F	1	4	F	4-12	PUSH RP	11RP 0101	1	12	S W W	4-15
CMP REG	1011 1SSS	1	4	F	4-11	RAL	17	1	4	F	4-12
CMP M	BE	1	7	F R	4-11	RAR	1F	1	4	F	4-12
CNC LABEL	D4 addr	3	9/18	S R/S R R W W	4-14	RC	D8	1	6/12	S/S R R	4-14
CNZ LABEL	C4 addr	3	9/18	S R/S R R W W	4-14	RET	C9	1	10	F R R	4-14
CP LABEL	F4 addr	3	9/18	S R/S R R W W	4-14	RIM	20	1	4	F	4-17
CPE LABEL	EC addr	3	9/18	S R/S R R W W	4-14	RLC	07	1	4	F	4-11
CPI DATA	FE data	2	7	F R	4-11	RM	F8	1	6/12	S/S R R	4-14
CPO LABEL	E4 addr	3	9/18	S R/S R R W W	4-14	RNC	D0	1	6/12	S/S R R	4-14
CZ LABEL	CC addr	3	9/18	S R/S R R W W	4-14	RNZ	C0	1	6/12	S/S R R	4-14
DAA	27	1	4	F	4-9	RP	F0	1	6/12	S/S R R	4-14
DAD RP	00RP 1001	1	10	F B B	4-9	RPE	E8	1	6/12	S/S R R	4-14
DCR REG	00SS S101	1	4	F	4-8	RPO	E0	1	6/12	S/S R R	4-14
DCR M	35	1	10	F R W	4-8	RRC	0F	1	4	F	4-12
DCX RP	00RP 1011	1	6	S	4-9	RST N	11XX X111	1	12	S R R	4-14
DI	F3	1	4	F	4-17	RZ	C8	1	6/12	S/S R R	4-14
EI	FB	1	4	F	4-17	SBB REG	1001 1SSS	1	4	F	4-7
HLT	76	1	5	F B	4-17	SBB M	9E	1	7	F R	4-8
IN PORT	DB data	2	10	F R I	4-16	SBI DATA	DE data	2	7	F R	4-8
INR REG	00SS S100	1	4	F	4-8	SHLD ADDR	22 addr	3	16	F R R W W	4-5
INR M	34	1	10	F R W	4-8	SIM	30	1	4	F	4-18
INX RP	00RP 0011	1	6	S	4-9	SPHL	F9	1	6	S	4-16
JC LABEL	DA addr	3	7/10	F R/F R R	4-13	STA ADDR	32 addr	3	13	F R R W	4-5
JM LABEL	FA addr	3	7/10	F R/F R R	4-13	STAX RP	000X 0010	1	7	FW	4-6
JMP LABEL	C3 addr	3	10	F R R	4-13	STC	37	1	4	F	4-12
JNC LABEL	D2 addr	3	7/10	F R/F R R	4-13	SUB REG	1001 0SSS	1	4	F	4-7
JNZ LABEL	C2 addr	3	7/10	F R/F R R	4-13	SUB M	96	1	7	F R	4-7
JP LABEL	F2 addr	3	7/10	F R/F R R	4-13	SUI DATA	D6 data	2	7	F R	4-7
JPE LABEL	EA addr	3	7/10	F R/F R R	4-13	XCHG	EB	1	4	F	4-6
JPO LABEL	E2 addr	3	7/10	F R/F R R	4-13	XRA REG	1010 1SSS	1	4	F	4-10
JZ LABEL	CA addr	3	7/10	F R/F R R	4-13	XRA M	AE	1	7	F R	4-10
LDA ADDR	3A addr	3	13	F R R R	4-5	XRI DATA	EE data	2	7	F R	4-10
LDAX RP	000X 1010	1	7	F R	4-5	XTHL	E3	1	16	F R R W W	4-16
LHLD ADDR	2A addr	3	16	F R R R R	4-5						

## Machine cycle types:

F Four clock period instr fetch  
 S Six clock period instr fetch  
 R Memory read  
 I I/O read  
 W Memory write  
 O I/O write  
 B Bus idle

X Variable or optional binary digit

DDD Binary digits identifying a destination register } B = 000, C = 001, D = 010 Memory = 110

SSS Binary digits identifying a source register } E = 011, H = 100, L = 101 A = 111

RP Register Pair { BC = 00, HL = 10  
DE = 01, SP = 11

## 8085A CPU INSTRUCTIONS IN OPERATION CODE SEQUENCE

Table 4-2

OP CODE	MNEMONIC	OP CODE	MNEMONIC	OP CODE	MNEMONIC	OP CODE	MNEMONIC	OP CODE	MNEMONIC	OP CODE	MNEMONIC
00	NOP	2B	DCX H	56	MOV D,M	81	ADD C	AC	XRA H	D7	RST 2
01	LXI B,D16	2C	INR L	57	MOV D,A	82	ADD D	AD	XRA L	D8	RC
02	STAX B	2D	DCR L	58	MOV E,B	83	ADD E	AE	XRA M	D9	—
03	INX B	2E	MVI L,D8	59	MOV E,C	84	ADD H	AF	XRA A	DA	JC Adr
04	INR B	2F	CMA	5A	MOV E,D	85	ADD L	B0	ORA B	DB	IN D8
05	DCR B	30	SIM	5B	MOV E,E	86	ADD M	B1	ORA C	DC	CC Adr
06	MVI B,D8	31	LXI SP,D16	5C	MOV E,H	87	ADD A	B2	ORA D	DD	—
07	RLC	32	STA Adr	5D	MOV E,L	88	ADC B	B3	ORA E	DE	SBI D8
08	—	33	INX SP	5E	MOV E,M	89	ADC C	B4	ORA H	DF	RST 3
09	DAD B	34	INR M	5F	MOV E,A	8A	ADC D	B5	ORA L	E0	RPO
0A	LDAX B	35	DCR M	60	MOV H,B	8B	ADC E	B6	ORA M	E1	POP H
0B	DCX B	36	MVI M,D8	61	MOV H,C	8C	ADC H	B7	ORA A	E2	JPO Adr
0C	INR C	37	STC	62	MOV H,D	8D	ADC L	B8	CMP B	E3	XTHL
0D	DCR C	38	—	63	MOV H,E	8E	ADC M	B9	CMP C	E4	CPO Adr
0E	MVI C,D8	39	DAD SP	64	MOV H,H	8F	ADC A	BA	CMP D	E5	PUSH H
0F	RRC	3A	LDA Adr	65	MOV H,L	90	SUB B	BB	CMP E	E6	ANI D8
10	—	3B	DCX SP	66	MOV H,M	91	SUB C	BC	CMP H	E7	RST 4
11	LXI D,D16	3C	INR A	67	MOV H,A	92	SUB D	BD	CMP L	E8	RPE
12	STAX D	3D	DCR A	68	MOV L,B	93	SUB E	BE	CMP M	E9	PCHL
13	INX D	3E	MVI A,D8	69	MOV L,C	94	SUB H	BF	CMP A	EA	JPE Adr
14	INR D	3F	CMC	6A	MOV L,D	95	SUB L	C0	RNZ	EB	XCHG
15	DCR D	40	MOV B,B	6B	MOV L,E	96	SUB M	C1	POP B	EC	CPE Adr
16	MVI D,D8	41	MOV B,C	6C	MOV L,H	97	SUB A	C2	JNZ Adr	ED	—
17	RAL	42	MOV B,D	6D	MOV L,L	98	SBB B	C3	JMP Adr	EE	XRI D8
18	—	43	MOV B,E	6E	MOV L,M	99	SBB C	C4	CNZ Adr	EF	RST 5
19	DAD D	44	MOV B,H	6F	MOV L,A	9A	SBB D	C5	PUSH B	F0	RP
1A	LDAX D	45	MOV B,L	70	MOV M,B	9B	SBB E	C6	ADI D8	F1	POP PSW
1B	DCX D	46	MOV B,M	71	MOV M,C	9C	SBB H	C7	RST 0	F2	JP Adr
1C	INR E	47	MOV B,A	72	MOV M,D	9D	SBB L	C8	RZ	F3	DI
1D	DCR E	48	MOV C,B	73	MOV M,E	9E	SBB M	C9	RET Adr	F4	CP Adr
1E	MVI E,D8	49	MOV C,C	74	MOV M,H	9F	SBB A	CA	JZ	F5	PUSH PSW
1F	RAR	4A	MOV C,D	75	MOV M,L	A0	ANA B	CB	—	F6	ORI D8
20	RIM	4B	MOV C,E	76	HLT	A1	ANA C	CC	CZ Adr	F7	RST 6
21	LXI H,D16	4C	MOV C,H	77	MOV M,A	A2	ANA D	CD	CALL Adr	F8	RM
22	SHLD Adr	4D	MOV C,L	78	MOV A,B	A3	ANA E	CE	ACI D8	F9	SPHL
23	INX H	4E	MOV C,M	79	MOV A,C	A4	ANA H	CF	RST 1	FA	JM Adr
24	INR H	4F	MOV C,A	7A	MOV A,D	A5	ANA L	D0	RNC	FB	EI
25	DCR H	50	MOV D,B	7B	MOV A,E	A6	ANA M	D1	POP D	FC	CM Adr
26	MVI H,D8	51	MOV D,C	7C	MOV A,H	A7	ANA A	D2	JNC Adr	FD	—
27	DAA	52	MOV D,D	7D	MOV A,L	A8	XRA B	D3	OUT D8	FE	CPI D8
28	—	53	MOV D,E	7E	MOV A,M	A9	XRA C	D4	CNC Adr	FF	RST 7
29	DAD H	54	MOV D,H	7F	MOV A,A	AA	XRA D	D5	PUSH D		
2A	LHLD Adr	55	MOV D,L	80	ADD B	AB	XRA E	D6	SUI D8		

D8 = constant, or logical/arithmetic expression that evaluates to an 8 bit data quantity.

Adr = 16-bit address.

D16 = constant, or logical/arithmetic expression that evaluates to a 16 bit data quantity.

**8085A INSTRUCTION SET SUMMARY BY FUNCTIONAL GROUPING**  
**Table 4-3**

Mnemonic	Description	Instruction Code(1)								Clock(2)
		D <sub>7</sub>	D <sub>6</sub>	D <sub>5</sub>	D <sub>4</sub>	D <sub>3</sub>	D <sub>2</sub>	D <sub>1</sub>	D <sub>0</sub>	Cycles
MOVE, LOAD, AND STORE										
MOVr1 r2	Move register to register	0	1	D	D	D	S	S	S	4
MOV M.r	Move register to memory	0	1	1	1	0	S	S	S	7
MOV r.M	Move memory to register	0	1	D	D	D	1	1	0	7
MVI r	Move immediate register	0	0	D	D	D	1	1	0	7
MVI M	Move immediate memory	0	0	1	1	0	1	1	0	10
LXI B	Load immediate register Pair B & C	0	0	0	0	0	0	0	1	10
LXI D	Load immediate register Pair D & E	0	0	0	1	0	0	0	1	10
LXI H	Load immediate register Pair H & L	0	0	1	0	0	0	0	1	10
LXI SP	Load immediate stack pointer	0	0	1	1	0	0	0	1	10
STAX B	Store A indirect	0	0	0	0	0	0	1	0	7
STAX D	Store A indirect	0	0	0	1	0	0	1	0	7
LDAX B	Load A indirect	0	0	0	0	1	0	1	0	7
LDAX D	Load A indirect	0	0	0	1	1	0	1	0	7
STA	Store A direct	0	0	1	1	0	0	1	0	13
LDA	Load A direct	0	0	1	1	1	0	1	0	13
SHLD	Store H & L direct	0	0	1	0	0	0	1	0	16
LHLD	Load H & L direct	0	0	1	0	1	0	1	0	16
XCHG	Exchange D & E, H & L Registers	1	1	1	0	1	0	1	1	4
STACK OPS										
PUSH B	Push register Pair B & C on stack	1	1	0	0	0	1	0	1	12
PUSH D	Push register Pair D & E on stack	1	1	0	1	0	1	0	1	12
PUSH H	Push register Pair H & L on stack	1	1	1	0	0	1	0	1	12
PUSH PSW	Push A and Flags on stack	1	1	1	1	0	1	0	1	12
POP B	Pop register Pair B & C off stack	1	1	0	0	0	0	0	1	10
POP D	Pop register Pair D & E off stack	1	1	0	1	0	0	0	1	10
POP H	Pop register Pair H & L off stack	1	1	1	0	0	0	0	1	10
POP PSW	Pop A and Flags off stack	1	1	1	1	0	0	0	1	10
XTHL	Exchange top of stack, H & L	1	1	1	0	0	0	1	1	16
SPHL	H & L to stack pointer	1	1	1	1	1	0	0	1	6
JUMP										
JMP	Jump unconditional	1	1	0	0	0	0	1	1	10
JC	Jump on carry	1	1	0	1	1	0	1	0	7/10
JNC	Jump on no carry	1	1	0	1	0	0	1	0	7/10
JZ	Jump on zero	1	1	0	0	1	0	1	0	7/10
JNZ	Jump on no zero	1	1	0	0	0	0	1	0	7/10
JP	Jump on positive	1	1	1	1	0	0	1	0	7/10
JM	Jump on minus	1	1	1	1	1	0	1	0	7/10
JPE	Jump on parity even	1	1	1	0	1	0	1	0	7/10
JPO	Jump on parity odd	1	1	1	0	0	0	1	0	7/10
PCHL	H & L to program counter	1	1	1	0	1	0	0	1	6
CALL										
CALL	Call unconditional	1	1	0	0	1	1	0	1	18
CC	Call on carry	1	1	0	1	1	1	0	0	9/18
Instruction Code(1)										
RETURN										
CNC	Call on no carry	1	1	0	1	0	1	0	0	9/18
CZ	Call on zero	1	1	0	0	1	1	0	0	9/18
CNZ	Call on no zero	1	1	0	0	0	1	0	0	9/18
CP	Call on positive	1	1	1	1	0	1	0	0	9/18
CM	Call on minus	1	1	1	1	1	1	0	0	9/18
CPE	Call on parity even	1	1	1	0	1	1	0	0	9/18
CPO	Call on parity odd	1	1	1	0	0	1	0	0	9/18
RETURN										
RET	Return	1	1	0	0	1	0	0	1	10
RC	Return on carry	1	1	0	1	1	0	0	0	6/12
RNC	Return on no carry	1	1	0	1	0	0	0	0	6/12
RZ	Return on zero	1	1	0	0	1	0	0	0	6/12
RNZ	Return on no zero	1	1	0	0	0	0	0	0	6/12
RP	Return on positive	1	1	1	1	0	0	0	0	6/12
RM	Return on minus	1	1	1	1	1	0	0	0	6/12
RPE	Return on parity even	1	1	1	0	1	0	0	0	6/12
RPO	Return on parity odd	1	1	1	0	0	0	0	0	6/12
RESTART										
RST	Restart	1	1	A	A	A	1	1	1	12
INPUT/OUTPUT										
IN	Input	1	1	0	1	1	0	1	1	10
OUT	Output	1	1	0	1	0	0	1	1	10
INCREMENT AND DECREMENT										
INR r	Increment register	0	0	D	D	D	1	0	0	4
DCR r	Decrement register	0	0	D	D	D	1	0	1	4
INR M	Increment memory	0	0	1	1	0	1	0	0	10
DCR M	Decrement memory	0	0	1	1	0	1	0	1	10
INX B	Increment B & C registers	0	0	0	0	0	0	1	1	6
INX D	Increment D & E registers	0	0	0	1	0	0	1	1	6
INX H	Increment H & L registers	0	0	1	0	0	0	1	1	6
INX SP	Increment stack pointer	0	0	1	1	0	0	1	1	6
DCX B	Decrement B & C	0	0	0	0	1	0	1	1	6
DCX D	Decrement D & E	0	0	0	1	1	0	1	1	6
DCX H	Decrement H & L	0	0	1	0	1	0	1	1	6
DCX SP	Decrement stack pointer	0	0	1	1	1	0	1	1	6
ADD										
ADD r	Add register to A	1	0	0	0	0	S	S	S	4
ADC r	Add register to A with carry	1	0	0	0	1	S	S	S	4
ADD M	Add memory to A	1	0	0	0	0	1	1	0	7
ADC M	Add memory to A with carry	1	0	0	0	1	1	1	0	7
ADI	Add immediate to A	1	1	0	0	0	1	1	0	7
ACI	Add immediate to A with carry	1	1	0	0	1	1	1	0	7
DAD B	Add B & C to H & L	0	0	0	0	1	0	0	1	10
DAD D	Add D & E to H & L	0	0	0	1	1	0	0	1	10
DAD H	Add H & L to H & L	0	0	1	0	1	0	0	1	10
DAD SP	Add stack pointer to H & L	0	0	1	1	1	0	0	1	10
SUBTRACT										
SUB r	Subtract register from A	1	0	0	1	0	S	S	S	4

TABLE 4-3

Mnemonic	Description	Instruction Code(1)								Clock(2)
		D <sub>7</sub>	D <sub>6</sub>	D <sub>5</sub>	D <sub>4</sub>	D <sub>3</sub>	D <sub>2</sub>	D <sub>1</sub>	D <sub>0</sub>	
SBB r	Subtract register from A with borrow	1	0	0	1	1	S	S	S	4
SUB M	Subtract memory from A	1	0	0	1	0	1	1	0	7
SBB M	Subtract memory from A with borrow	1	0	0	1	1	1	1	0	7
SUI	Subtract immediate from A	1	1	0	1	0	1	1	0	7
SBI	Subtract immediate from A with borrow	1	1	0	1	1	1	1	0	7
<b>LOGICAL</b>										
ANA r	And register with A	1	0	1	0	0	S	S	S	4
XRA r	Exclusive OR register with A	1	0	1	0	1	S	S	S	4
ORA r	OR register with A	1	0	1	1	0	S	S	S	4
CMP r	Compare register with A	1	0	1	1	1	S	S	S	4
ANA M	And memory with A	1	0	1	0	0	1	1	0	7
XRAM	Exclusive OR memory with A	1	0	1	0	1	1	1	0	7
ORAM	OR memory with A	1	0	1	1	0	1	1	0	7
CMPM	Compare memory with A	1	0	1	1	1	1	1	0	7
ANI	And immediate with A	1	1	1	0	0	1	1	0	7
XRI	Exclusive OR immediate with A	1	1	1	0	1	1	1	0	7

NOTES: 1. DDS or SSS: B 000, C 001, D 010, E 011, H 100, L 101, Memory 110, A 111.

2. Two possible cycle times. (6/12) indicate instruction cycles dependent on condition flags.

\*All mnemonics copyright © Intel Corporation 1977

Mnemonic	Description	Instruction Code(1)								Clock(2)
		D <sub>7</sub>	D <sub>6</sub>	D <sub>5</sub>	D <sub>4</sub>	D <sub>3</sub>	D <sub>2</sub>	D <sub>1</sub>	D <sub>0</sub>	
ORI	OR immediate with A	1	1	1	1	0	1	1	0	7
CPI	Compare immediate with A	1	1	1	1	1	1	1	0	7
<b>ROTATE</b>										
RLC	Rotate A left	0	0	0	0	0	1	1	1	4
RRC	Rotate A right	0	0	0	0	1	1	1	1	4
RAL	Rotate A left through carry	0	0	0	1	0	1	1	1	4
RAR	Rotate A right through carry	0	0	0	1	1	1	1	1	4
<b>SPECIALS</b>										
CMA	Complement A	0	0	1	0	1	1	1	1	4
STC	Set carry	0	0	1	1	0	1	1	1	4
CMC	Complement carry	0	0	1	1	1	1	1	1	4
DAA	Decimal adjust A	0	0	1	0	0	1	1	1	4
<b>CONTROL</b>										
EI	Enable Interrupts	1	1	1	1	1	0	1	1	4
DI	Disable Interrupt	1	1	1	1	0	0	1	1	4
NOP	No-operation	0	0	0	0	0	0	0	0	4
HLT	Halt	0	1	1	1	0	1	1	0	5
<b>NEW 8085A INSTRUCTIONS</b>										
RIM	Read Interrupt Mask	0	0	1	0	0	0	0	0	4
SIM	Set Interrupt Mask	0	0	1	1	0	0	0	0	4

## **MCS-85**

System Support  
Components

Peripherals

Static RAMs

ROMs/EPROMs

# **MCS 85**

# **MCS 85**

# **MCS 85**



# 8085A/8085A-2

## SINGLE CHIP 8-BIT N-CHANNEL MICROPROCESSORS

- **Single +5V Power Supply**
- **100% Software Compatible with 8080A**
- **1.3  $\mu$ s Instruction Cycle (8085A); 0.8  $\mu$ s (8085A-2)**
- **On-Chip Clock Generator (with External Crystal, LC or RC Network)**
- **On-Chip System Controller; Advanced Cycle Status Information Available for Large System Control**
- **Four Vectored Interrupt Inputs (One is non-Maskable) Plus an 8080A-compatible interrupt**
- **Serial In/Serial Out Port**
- **Decimal, Binary and Double Precision Arithmetic**
- **Direct Addressing Capability to 64k Bytes of Memory**

The Intel® 8085A is a complete 8 bit parallel Central Processing Unit (CPU). Its instruction set is 100% software compatible with the 8080A microprocessor, and it is designed to improve the present 8080A's performance by higher system speed. Its high level of system integration allows a minimum system of three IC's [8085A (CPU), 8156 (RAM/IO) and 8355/8755A (ROM/PROM/IO)] while maintaining total system expandability. The 8085A-2 is a faster version of the 8085A.

The 8085A incorporates all of the features that the 8224 (clock generator) and 8228 (system controller) provided for the 8080A, thereby offering a high level of system integration.

The 8085A uses a multiplexed data bus. The address is split between the 8 bit address bus and the 8 bit data bus. The on-chip address latches of 8155/8156/8355/8755A memory products allow a direct interface with the 8085A.

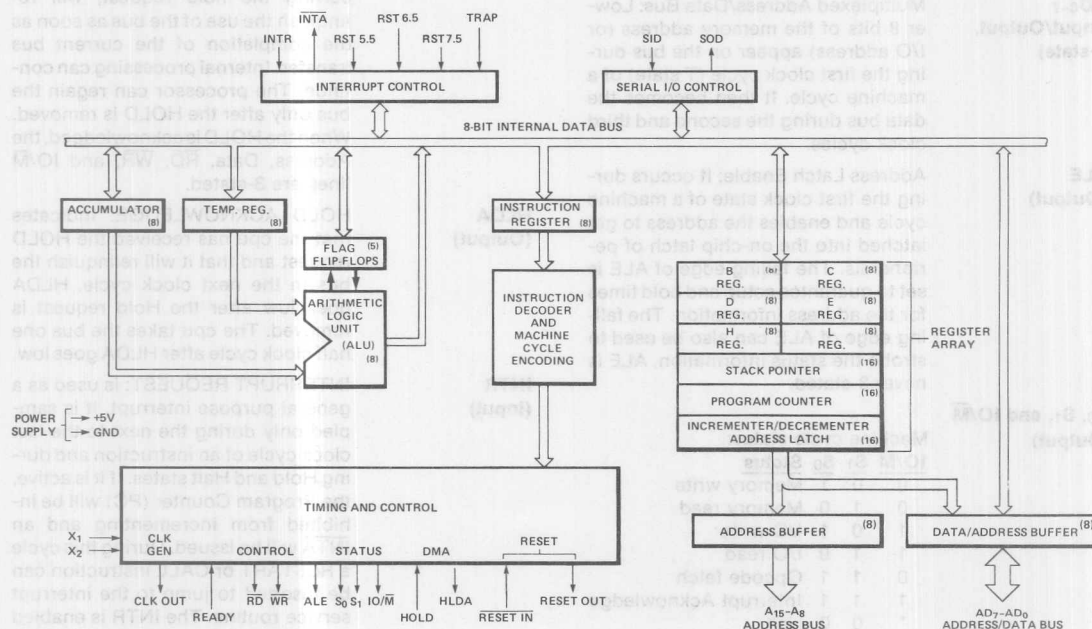


Figure 1. 8085A CPU Functional Block Diagram

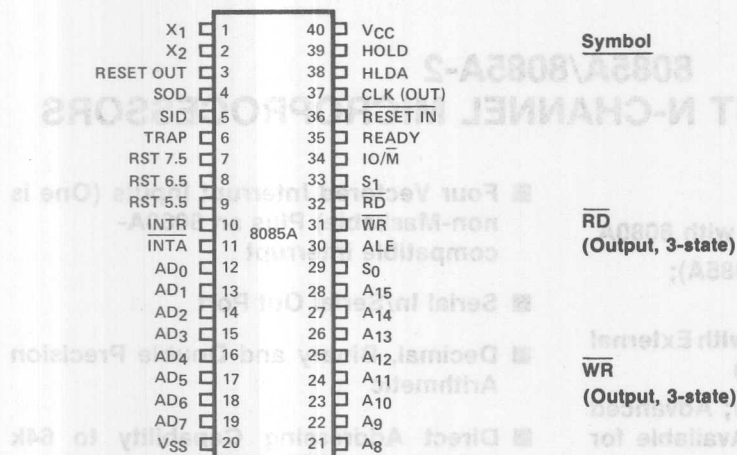


Figure 2. 8085A Pinout Diagram

## 8085A FUNCTIONAL PIN DEFINITION

The following describes the function of each pin:

### Symbol

### Function

**A<sub>8</sub>-A<sub>15</sub>**  
(Output, 3-state)

Address Bus: The most significant 8 bits of the memory address or the 8 bits of the I/O address, 3-stated during Hold and Halt modes and during RESET.

**AD<sub>0-7</sub>**  
(Input/Output, 3-state)

Multiplexed Address/Data Bus: Lower 8 bits of the memory address (or I/O address) appear on the bus during the first clock cycle (T state) of a machine cycle. It then becomes the data bus during the second and third clock cycles.

**ALE**  
(Output)

Address Latch Enable: It occurs during the first clock state of a machine cycle and enables the address to get latched into the on-chip latch of peripherals. The falling edge of ALE is set to guarantee setup and hold times for the address information. The falling edge of ALE can also be used to strobe the status information. ALE is never 3-stated.

**S<sub>0</sub>, S<sub>1</sub>, and IO/M**  
(Output)

Machine cycle status:

IO/M	S <sub>1</sub>	S <sub>0</sub>	Status
0	0	1	Memory write
0	1	0	Memory read
1	0	1	I/O write
1	1	0	I/O read
0	1	1	Opcode fetch
1	1	1	Interrupt Acknowledge
*	0	0	Halt
*	X	X	Hold
*	X	X	Reset

\* = 3-state (high impedance)

X = unspecified

### Symbol

### Function

S<sub>1</sub> can be used as an advanced R/W status. IO/M, S<sub>0</sub> and S<sub>1</sub> become valid at the beginning of a machine cycle and remain stable throughout the cycle. The falling edge of ALE may be used to latch the state of these lines.

**RD**  
(Output, 3-state)

READ control: A low level on  $\overline{RD}$  indicates the selected memory or I/O device is to be read and that the Data Bus is available for the data transfer, 3-stated during Hold and Halt modes and during RESET.

**WR**  
(Output, 3-state)

WRITE control: A low level on  $\overline{WR}$  indicates the data on the Data Bus is to be written into the selected memory or I/O location. Data is set up at the trailing edge of  $\overline{WR}$ . 3-stated during Hold and Halt modes and during RESET.

**READY**  
(Input)

If READY is high during a read or write cycle, it indicates that the memory or peripheral is ready to send or receive data. If READY is low, the cpu will wait an integral number of clock cycles for READY to go high before completing the read or write cycle.

**HOLD**  
(Input)

HOLD indicates that another master is requesting the use of the address and data buses. The cpu, upon receiving the hold request, will relinquish the use of the bus as soon as the completion of the current bus transfer. Internal processing can continue. The processor can regain the bus only after the HOLD is removed. When the HOLD is acknowledged, the Address, Data,  $\overline{RD}$ ,  $\overline{WR}$ , and IO/M lines are 3-stated.

**HLDA**  
(Output)

HOLD ACKNOWLEDGE: Indicates that the cpu has received the HOLD request and that it will relinquish the bus in the next clock cycle. HLDA goes low after the Hold request is removed. The cpu takes the bus one half clock cycle after HLDA goes low.

**INTR**  
(Input)

INTERRUPT REQUEST: is used as a general purpose interrupt. It is sampled only during the next to the last clock cycle of an instruction and during Hold and Halt states. If it is active, the Program Counter (PC) will be inhibited from incrementing and an INTA will be issued. During this cycle a RESTART or CALL instruction can be inserted to jump to the interrupt service routine. The INTR is enabled and disabled by software. It is disabled by Reset and immediately after an interrupt is accepted.

## 8085A FUNCTIONAL PIN DESCRIPTION (Continued)

Symbol	Function	Symbol	Function
<b>INTA</b> (Output)	<b>INTERRUPT ACKNOWLEDGE:</b> Is used instead of (and has the same timing as) $\overline{RD}$ during the Instruction cycle after an INTR is accepted. It can be used to activate the 8259 Interrupt chip or some other interrupt port.	<b>RESET OUT</b> (Output)	Schmitt-triggered input, allowing connection to an R-C network for power-on RESET delay. The cpu is held in the reset condition as long as RESET IN is applied.
<b>RST 5.5</b> <b>RST 6.5</b> <b>RST 7.5</b> (Inputs)	<b>RESTART INTERRUPTS:</b> These three inputs have the same timing as INTR except they cause an internal RESTART to be automatically inserted.  The priority of these interrupts is ordered as shown in Table 1. These interrupts have a higher priority than INTR. In addition, they may be individually masked out using the SIM instruction.	<b>X1, X2</b> (Input)	Indicates cpu is being reset. Can be used as a system reset. The signal is synchronized to the processor clock and lasts an integral number of clock periods.  X1 and X2 are connected to a crystal, LC, or RC network to drive the internal clock generator. X1 can also be an external clock input from a logic gate. The input frequency is divided by 2 to give the processor's internal operating frequency.
<b>TRAP</b> (Input)	Trap interrupt is a nonmaskable RESTART interrupt. It is recognized at the same time as INTR or RST 5.5-7.5. It is unaffected by any mask or Interrupt Enable. It has the highest priority of any interrupt. (See Table 1.)	<b>CLK</b> (Output)	Clock Output for use as a system clock. The period of CLK is twice the X1, X2 input period.
<b>RESET IN</b> (Input)	Sets the Program Counter to zero and resets the Interrupt Enable and HLDA flip-flops. The data and address buses and the control lines are 3-stated during RESET and because of the asynchronous nature of RESET, the processor's internal registers and flags may be altered by RESET with unpredictable results. RESET IN is a	<b>SID</b> (Input)	Serial input data line. The data on this line is loaded into accumulator bit 7 whenever a RIM instruction is executed.
		<b>SOD</b> (Output)	Serial output data line. The output SOD is set or reset as specified by the SIM instruction.
		<b>Vcc</b>	+5 volt supply.
		<b>Vss</b>	Ground Reference.

TABLE 1. INTERRUPT PRIORITY, RESTART ADDRESS, AND SENSITIVITY

Name	Priority	Address Branched To (1) When Interrupt Occurs	Type Trigger
TRAP	1	24H	Rising edge AND high level until sampled.
RST 7.5	2	3CH	Rising edge (latched).
RST 6.5	3	34H	High level until sampled.
RST 5.5	4	2CH	High level until sampled.
INTR	5	See Note (2).	High level until sampled.

## NOTES:

- (1) The processor pushes the PC on the stack before branching to the indicated address.
- (2) The address branched to depends on the instruction provided to the cpu when the interrupt is acknowledged.

## FUNCTIONAL DESCRIPTION

The 8085A is a complete 8-bit parallel central processor. It is designed with N-channel depletion loads and requires a single +5 volt supply. Its basic clock speed is 3 MHz (8085A) or 5 MHz (8085A-2), thus improving on the present 8080A's performance with higher system speed. Also it is designed to fit into a minimum system of three IC's: The cpu (8085A), a RAM/IO (8156), and a ROM or EPROM/IO chip (8355 or 8755A).

The 8085A has twelve addressable 8-bit registers. Four of them can function only as two 16-bit register pairs. Six others can be used interchangeably as 8-bit registers or as 16-bit register pairs. The 8085A register set is as follows:

Mnemonic	Register	Contents
ACC or A	Accumulator	8 bits
PC	Program Counter	16-bit address
BC,DE,HL	General-Purpose Registers; data pointer (HL)	8 bits x 6 or 16 bits x 3
SP	Stack Pointer	16-bit address
Flags or F	Flag Register	5 flags (8-bitspace)

The 8085A uses a multiplexed Data Bus. The address is split between the higher 8-bit Address Bus and the lower 8-bit Address/Data Bus. During the first T state (clock cycle) of a machine cycle the low order address is sent out on the Address/Data bus. These lower 8 bits may be latched externally by the Address Latch Enable signal (ALE). During the rest of the machine cycle the data bus is used for memory or I/O data.

The 8085A provides  $\overline{RD}$ ,  $\overline{WR}$ ,  $S_0$ ,  $S_1$ , and  $IO/\overline{M}$  signals for bus control. An Interrupt Acknowledge signal ( $\overline{INTA}$ ) is also provided. HOLD, READY, and all Interrupts are synchronized with the processor's internal clock. The 8085A also provides Serial Input Data (SID) and Serial Output Data (SOD) lines for simple serial interface.

In addition to these features, the 8085A has three maskable, vector interrupt pins and one nonmaskable TRAP interrupt.

## INTERRUPT AND SERIAL I/O

The 8085A has 5 interrupt inputs: INTR, RST 5.5, RST 6.5, RST 7.5, and TRAP. INTR is identical in function to the 8080A INT. Each of the three RESTART inputs, 5.5, 6.5, and 7.5, has a programmable mask. TRAP is also a RESTART interrupt but it is nonmaskable.

The three maskable interrupts cause the internal execution of RESTART (saving the program counter in the stack and branching to the RESTART address) if the interrupts are enabled and if the interrupt mask is not set. The non-maskable TRAP causes the internal execution of a RESTART vector independent of the state of the interrupt enable or masks. (See Table 1.)

There are two different types of inputs in the restart interrupts. RST 5.5 and RST 6.5 are *high level-sensitive* like INTR (and INT on the 8080) and are recognized with the same timing as INTR. RST 7.5 is *rising edge-sensitive*.

For RST 7.5, only a pulse is required to set an internal flip-flop which generates the internal interrupt request. (See Section 2.2.7.) The RST 7.5 request flip-flop remains

set until the request is serviced. Then it is reset automatically. This flip-flop may also be reset by using the SIM instruction or by issuing a RESET IN to the 8085A. The RST 7.5 internal flip-flop will be set by a pulse on the RST 7.5 pin even when the RST 7.5 interrupt is masked out.

The status of the three RST interrupt masks can only be affected by the SIM instruction and RESET IN. (See SIM, Chapter 4.)

The interrupts are arranged in a fixed priority that determines which interrupt is to be recognized if more than one is pending as follows: TRAP — highest priority, RST 7.5, RST 6.5, RST 5.5, INTR — lowest priority. This priority scheme does not take into account the priority of a routine that was started by a higher priority interrupt. RST 5.5 can interrupt an RST 7.5 routine if the interrupts are re-enabled before the end of the RST 7.5 routine.

The TRAP interrupt is useful for catastrophic events such as power failure or bus error. The TRAP input is recognized just as any other interrupt but has the highest priority. It is not affected by any flag or mask. The TRAP input is both *edge and level sensitive*. The TRAP input must go high and remain high until it is acknowledged. It will not be recognized again until it goes low, then high again. This avoids any false triggering due to noise or logic glitches. Figure 3 illustrates the TRAP interrupt request circuitry within the 8085A. Note that the servicing of any interrupt (TRAP, RST 7.5, RST 6.5, RST 5.5, INTR) disables all future interrupts (except TRAPs) until an EI instruction is executed.

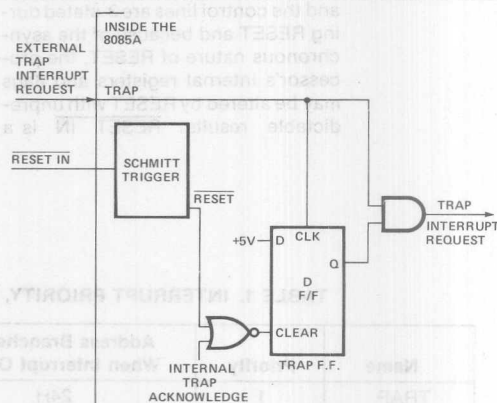


Figure 3. TRAP and RESET IN Circuit

The TRAP interrupt is special in that it disables interrupts, but preserves the previous interrupt enable status. Performing the first RIM instruction following a TRAP interrupt allows you to determine whether interrupts were enabled or disabled prior to the TRAP. All subsequent RIM instructions provide current interrupt enable status. Performing a RIM instruction following INTR, or RST 5.5-7.5 will provide current Interrupt Enable status, revealing that Interrupts are disabled. See the description of the RIM instruction in Chapter 4.

The serial I/O system is also controlled by the RIM and SIM instructions. SID is read by RIM, and SIM sets the SOD data.

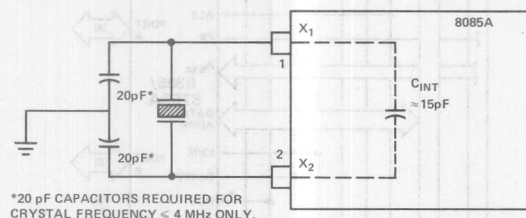
## DRIVING THE X<sub>1</sub> AND X<sub>2</sub> INPUTS

You may drive the clock inputs of the 8085A or 8085A-2 with a crystal, an LC tuned circuit, an RC network, or an external clock source. The driving frequency must be at least 1 MHz, and must be twice the desired internal clock frequency; hence, the 8085A is operated with a 6 MHz crystal (for 3 MHz clock), and the 8085A-2 can be operated with a 10 MHz crystal (for 5 MHz clock). If a crystal is used, it must have the following characteristics:

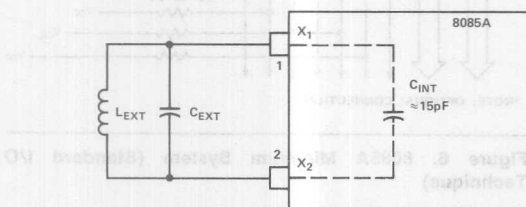
Parallel resonance at twice the clock frequency desired  
 $C_L$  (load capacitance)  $\leq 30$  pF  
 $C_s$  (shunt capacitance)  $\leq 7$  pF  
 $R_s$  (equivalent shunt resistance)  $\leq 75$  Ohms  
 Drive level: 10 mW  
 Frequency tolerance:  $\pm 0.005\%$  (suggested)

Note the use of the 20 pF capacitors between X<sub>1</sub>, X<sub>2</sub> and ground. These capacitors are required with crystal frequencies below 4 MHz to assure oscillator startup at the correct frequency. A parallel-resonant LC circuit may be used as the frequency-determining network for the 8085A, providing that its frequency tolerance of approximately  $\pm 10\%$  is acceptable. The components are chosen from the formula:

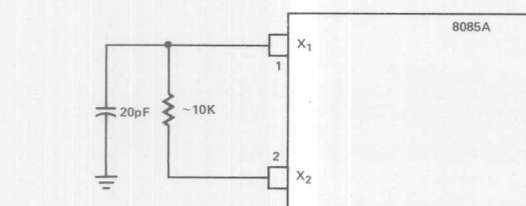
$$f = \frac{1}{2\pi\sqrt{L(C_{\text{ext}} + C_{\text{int}})}}$$



**A. Quartz Crystal Clock Driver**



**B. LC Tuned Circuit Clock Driver**



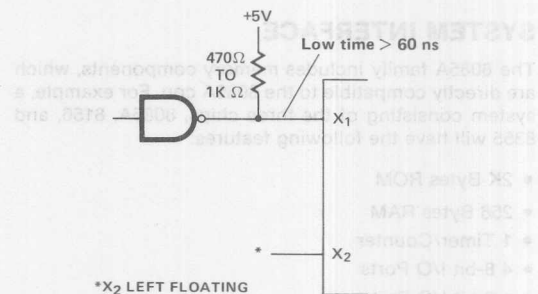
**C. RC Circuit Clock Driver**

To minimize variations in frequency, it is recommended that you choose a value for  $C_{\text{ext}}$  that is at least twice that of  $C_{\text{int}}$ , or 30 pF. The use of an LC circuit is not recommended for frequencies higher than approximately 5 MHz.

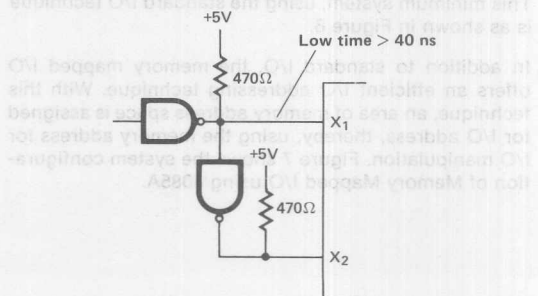
An RC circuit may be used as the frequency-determining network for the 8085A if maintaining a precise clock frequency is of no importance. Variations in the on-chip timing generation can cause a wide variation in frequency when using the RC mode. Its advantage is its low component cost. The driving frequency generated by the circuit shown is approximately 3 MHz. It is not recommended that frequencies greatly higher or lower than this be attempted.

Figure 4 shows the recommended clock driver circuits. Note in D and E that pullup resistors are required to assure that the high level voltage of the input is at least 4 V.

For driving frequencies up to and including 6 MHz you may supply the driving signal to X<sub>1</sub> and leave X<sub>2</sub> open-circuited (Figure 4D). If the driving frequency is from 6 MHz to 10 MHz, stability of the clock generator will be improved by driving both X<sub>1</sub> and X<sub>2</sub> with a push-pull source (Figure 4E). To prevent self-oscillation of the 8085A, be sure that X<sub>2</sub> is not coupled back to X<sub>1</sub> through the driving circuit.



**D. 1-6 MHz Input Frequency External Clock Driver Circuit**



**E. 1-10 MHz Input Frequency External Clock Driver Circuit**

**Figure 4. Clock Driver Circuits**

## GENERATING AN 8085A WAIT STATE

If your system requirements are such that slow memories or peripheral devices are being used, the circuit shown in Figure 5 may be used to insert one WAIT state in each 8085A machine cycle.

The D flip-flops should be chosen so that

- CLK is rising edge-triggered
- CLEAR is low-level active.

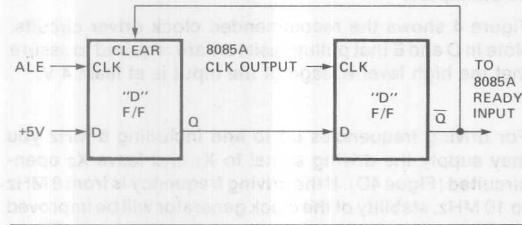


Figure 5. Generation of a Wait State for 8085A CPU

As in the 8080, the READY line is used to extend the read and write pulse lengths so that the 8085A can be used with slow memory. HOLD causes the cpu to relinquish the bus when it is through with it by floating the Address and Data Buses.

## SYSTEM INTERFACE

The 8085A family includes memory components, which are directly compatible to the 8085A cpu. For example, a system consisting of the three chips, 8085A, 8156, and 8355 will have the following features:

- 2K Bytes ROM
- 256 Bytes RAM
- 1 Timer/Counter
- 4 8-bit I/O Ports
- 1 16-bit I/O Port
- 4 Interrupt Levels
- Serial In/Serial Out Ports

This minimum system, using the standard I/O technique is as shown in Figure 6.

In addition to standard I/O, the memory mapped I/O offers an efficient I/O addressing technique. With this technique, an area of memory address space is assigned for I/O address, thereby, using the memory address for I/O manipulation. Figure 7 shows the system configuration of Memory Mapped I/O using 8085A.

The 8085A cpu can also interface with the standard memory that does *not* have the multiplexed address/data bus. It will require a simple 8212 (8-bit latch) as shown in Figure 8.

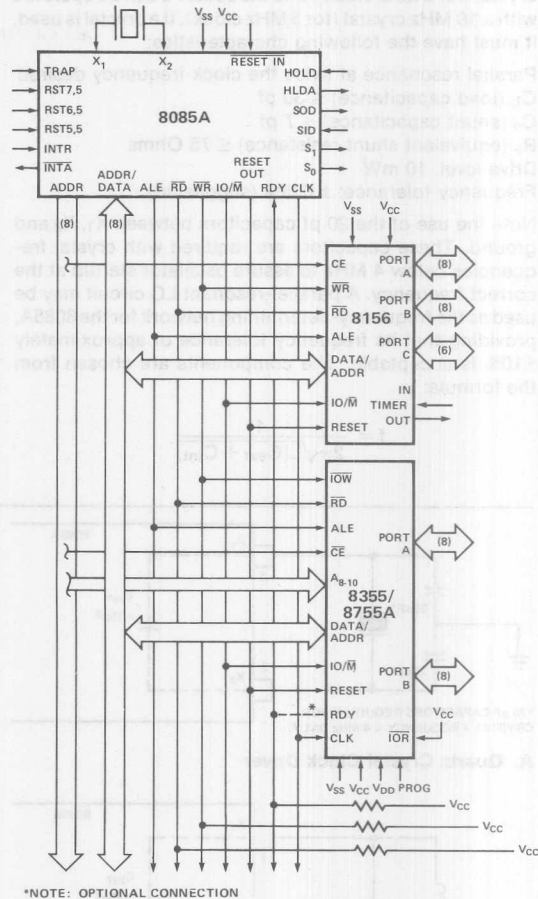
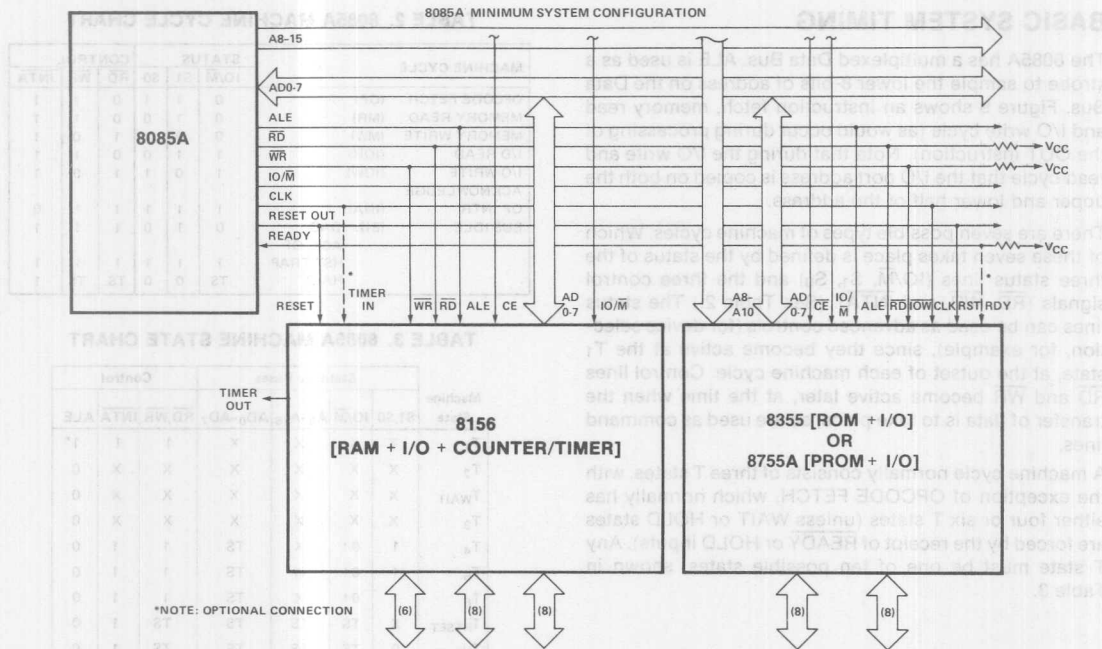
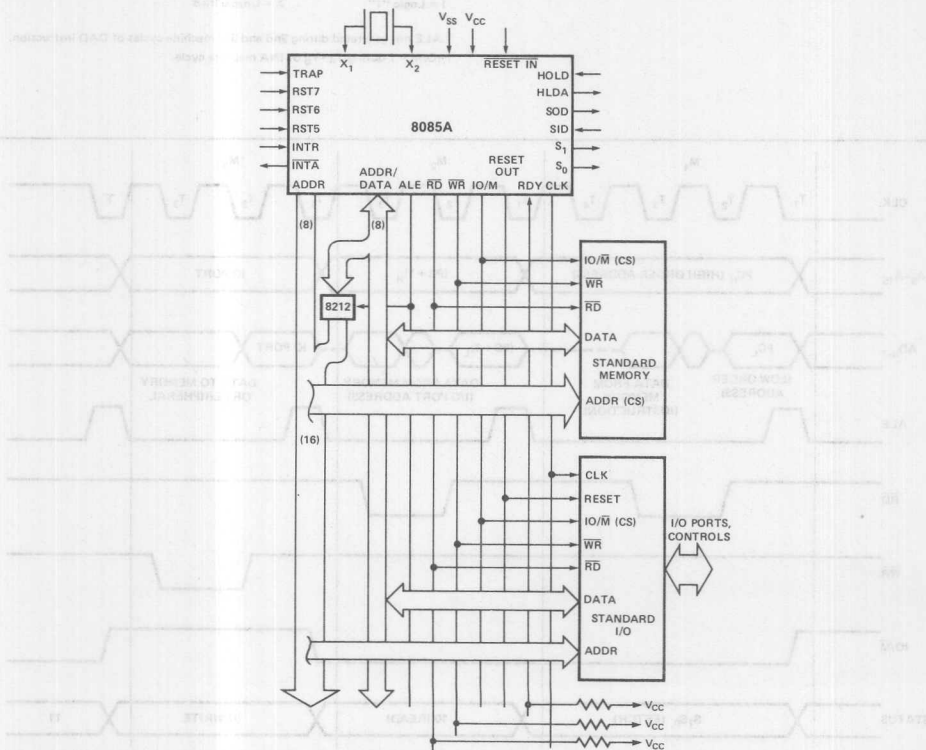


Figure 6. 8085A Minimum System (Standard I/O Technique)

## 8085A/8085A-2



**Figure 7. MCS-85™ Minimum System (Memory Mapped I/O)**



**Figure 8. MCS-85™ System (Using Standard Memories)**

## BASIC SYSTEM TIMING

The 8085A has a multiplexed Data Bus. ALE is used as a strobe to sample the lower 8-bits of address on the Data Bus. Figure 9 shows an instruction fetch, memory read and I/O write cycle (as would occur during processing of the OUT instruction). Note that during the I/O write and read cycle that the I/O port address is copied on both the upper and lower half of the address.

There are seven possible types of machine cycles. Which of these seven takes place is defined by the status of the three status lines ( $\overline{IO/\overline{M}}$ ,  $S_1$ ,  $S_0$ ) and the three control signals ( $\overline{RD}$ ,  $\overline{WR}$ , and  $\overline{INTA}$ ). (See Table 2.) The status lines can be used as advanced controls (for device selection, for example), since they become active at the  $T_1$  state, at the outset of each machine cycle. Control lines  $\overline{RD}$  and  $\overline{WR}$  become active later, at the time when the transfer of data is to take place, so are used as command lines.

A machine cycle normally consists of three T states, with the exception of OP CODE FETCH, which normally has either four or six T states (unless WAIT or HOLD states are forced by the receipt of  $\overline{READY}$  or  $\overline{HOLD}$  inputs). Any T state must be one of ten possible states, shown in Table 3.

TABLE 2. 8085A MACHINE CYCLE CHART

MACHINE CYCLE	STATUS			CONTROL		
	$\overline{IO/\overline{M}}$	$S_1$	$S_0$	$\overline{RD}$	$\overline{WR}$	$\overline{INTA}$
OPCODE FETCH (OF)	0	1	1	0	1	1
MEMORY READ (MR)	0	1	0	0	1	1
MEMORY WRITE (MW)	0	0	1	1	0	1
I/O READ (IOR)	1	1	0	0	1	1
I/O WRITE (IOW)	1	0	1	1	0	1
ACKNOWLEDGE OF INTR (INA)	1	1	1	1	1	0
BUS IDLE (BI): DAD	0	1	0	1	1	1
ACK. OF RST, TRAP	1	1	1	1	1	1
HALT	TS	0	0	TS	TS	1

TABLE 3. 8085A MACHINE STATE CHART

Machine State	Status & Buses				Control		
	$S_1, S_0$	$\overline{IO/\overline{M}}$	$A_8-A_{15}$	$AD_0-AD_7$	$\overline{RD}, \overline{WR}$	$\overline{INTA}$	ALE
$T_1$	X	X	X	X	1	1	1*
$T_2$	X	X	X	X	X	X	0
$T_{WAIT}$	X	X	X	X	X	X	0
$T_3$	X	X	X	X	X	X	0
$T_4$	1	0†	X	TS	1	1	0
$T_5$	1	0†	X	TS	1	1	0
$T_6$	1	0†	X	TS	1	1	0
$T_{RESET}$	X	TS	TS	TS	TS	1	0
$T_{HALT}$	0	TS	TS	TS	TS	1	0
$T_{HOLD}$	X	TS	TS	TS	TS	1	0

0 = Logic "0"  
1 = Logic "1"

TS = High Impedance  
X = Unspecified

\* ALE not generated during 2nd and 3rd machine cycles of DAD instruction.  
†  $\overline{IO/\overline{M}} = 1$  during  $T_4-T_6$  of INA machine cycle.

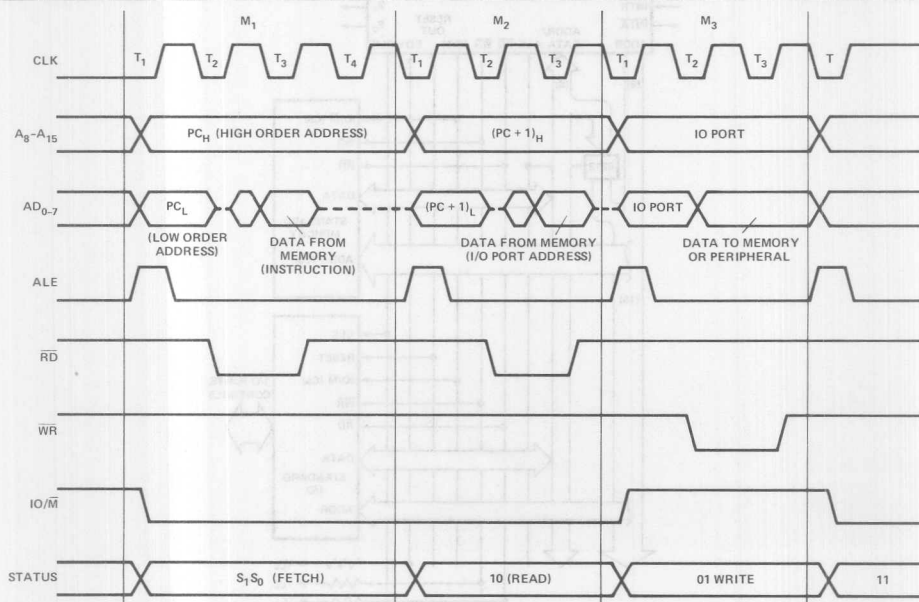


Figure 9. 8085A Basic System Timing

TABLE 4. ABSOLUTE MAXIMUM RATINGS\*

Ambient Temperature Under Bias. . . . .	0°C to 70°C
Storage Temperature . . . . .	-65°C to +150°C
Voltage on Any Pin . . . . .	
With Respect to Ground . . . . .	-0.5V to +7V
Power Dissipation . . . . .	1.5 Watt

## \*COMMENT

Stresses above those listed under "Absolute Maximum Ratings" may cause permanent damage to the device. This is a stress rating only and functional operation of the device at these or any other conditions above those indicated in the operational sections of this specification is not implied. Exposure to absolute maximum rating conditions for extended periods may affect device reliability.

TABLE 5. D.C. CHARACTERISTICS

( $T_A = 0^\circ\text{C}$  to  $70^\circ\text{C}$ ;  $V_{CC} = 5\text{V} \pm 5\%$ ;  $V_{SS} = 0\text{V}$ ; unless otherwise specified)

Symbol	Parameter	Min.	Max.	Units	Test Conditions
$V_{IL}$	Input Low Voltage	-0.5	+0.8	V	
$V_{IH}$	Input High Voltage	2.0	$V_{CC} + 0.5$	V	
$V_{OL}$	Output Low Voltage		0.45	V	$I_{OL} = 2\text{mA}$
$V_{OH}$	Output High Voltage	2.4		V	$I_{OH} = -400\mu\text{A}$
$I_{CC}$	Power Supply Current		170	mA	
$I_{IL}$	Input Leakage		$\pm 10$	$\mu\text{A}$	$V_{in} = V_{CC}$
$I_{LO}$	Output Leakage		$\pm 10$	$\mu\text{A}$	$0.45\text{V} \leq V_{out} \leq V_{CC}$
$V_{ILR}$	Input Low Level, RESET	-0.5	+0.8	V	
$V_{IHR}$	Input High Level, RESET	2.4	$V_{CC} + 0.5$	V	
$V_{HY}$	Hysteresis, RESET	0.25		V	

## 8085A/8085A-2

Symbol	Parameter	8085A <sup>[2]</sup>		8085A-2 <sup>[2]</sup> (Preliminary)		Units
		Min.	Max.	Min.	Max.	
t <sub>CYC</sub>	CLK Cycle Period	320	2000	200	2000	ns
t <sub>1</sub>	CLK Low Time	80		40		ns
t <sub>2</sub>	CLK High Time	120		70		ns
t <sub>r,tf</sub>	CLK Rise and Fall Time		30		30	ns
t <sub>XKR</sub>	X <sub>1</sub> Rising to CLK Rising	30	120	30	100	ns
t <sub>XKF</sub>	X <sub>1</sub> Rising to CLK Falling	30	150	30	110	ns
t <sub>AC</sub>	A <sub>8-15</sub> Valid to Leading Edge of Control <sup>[1]</sup>	270		115		ns
t <sub>ACL</sub>	A <sub>0-7</sub> Valid to Leading Edge of Control	240		115		ns
t <sub>AD</sub>	A <sub>0-15</sub> Valid to Valid Data In		575		350	ns
t <sub>AFR</sub>	Address Float After Leading Edge of READ (INTA)		0		0	ns
t <sub>AL</sub>	A <sub>8-15</sub> Valid Before Trailing Edge of ALE <sup>[1]</sup>	115		50		ns
t <sub>ALL</sub>	A <sub>0-7</sub> Valid Before Trailing Edge of ALE	90		50		ns
t <sub>ARY</sub>	READY Valid from Address Valid		220		100	ns
t <sub>CA</sub>	Address (A <sub>8-A15</sub> ) Valid After Control	120		60		ns
t <sub>CC</sub>	Width of Control Low ( $\overline{RD}$ , $\overline{WR}$ , INTA) Edge of ALE	400		230		ns
t <sub>CL</sub>	Trailing Edge of Control to Leading Edge of ALE	50		25		ns
t <sub>DW</sub>	Data Valid to Trailing Edge of $\overline{WRITE}$	420		230		ns
t <sub>HABE</sub>	HLDA to Bus Enable		210		150	ns
t <sub>HABF</sub>	Bus Float After HLDA		210		150	ns
t <sub>HACK</sub>	HLDA Valid to Trailing Edge of CLK	110		40		ns
t <sub>HDH</sub>	HOLD Hold Time	0		0		ns
t <sub>HDS</sub>	HOLD Setup Time to Trailing Edge of CLK	170		120		ns
t <sub>INH</sub>	INTR Hold Time	0		0		ns
t <sub>INS</sub>	INTR, RST, and TRAP Setup Time to Falling Edge of CLK	160		150		ns
t <sub>LA</sub>	Address Hold Time After ALE	100		50		ns
t <sub>LC</sub>	Trailing Edge of ALE to Leading Edge of Control	130		60		ns
t <sub>LCK</sub>	ALE Low During CLK High	100		50		ns
t <sub>LDR</sub>	ALE to Valid Data During Read		460		270	ns
t <sub>LDW</sub>	ALE to Valid Data During Write		200		120	ns
t <sub>LL</sub>	ALE Width	140		80		ns
t <sub>LRV</sub>	ALE to READY Stable		110		30	ns

TABLE 6. A.C. CHARACTERISTICS (Cont.)

Symbol	Parameter	8085A <sup>[2]</sup>		8085A-2 <sup>[2]</sup> (Preliminary)		Units
		Min.	Max.	Min.	Max.	
trAE	Trailing Edge of $\overline{\text{READ}}$ to Re-Enabling of Address	150		90		ns
trD	$\overline{\text{READ}}$ (or $\overline{\text{INTA}}$ ) to Valid Data		300		150	ns
trV	Control Trailing Edge to Leading Edge of Next Control	400		220		ns
trDH	Data Hold Time After $\overline{\text{READ}}$ $\overline{\text{INTA}}$ <sup>[7]</sup>	0		0		ns
trYH	READY Hold Time	0		0		ns
trYS	READY Setup Time to Leading Edge of CLK	110		100		ns
tWD	Data Valid After Trailing Edge of $\overline{\text{WRITE}}$	100		60		ns
tWDL	LEADING Edge of $\overline{\text{WRITE}}$ to Data Valid		40		20	ns

## Notes:

1. A<sub>8</sub>-A<sub>15</sub> address Specs apply to  $\overline{\text{IO}/\overline{\text{M}}}$ , S<sub>0</sub>, and S<sub>1</sub> except A<sub>8</sub>-A<sub>15</sub> are undefined during T<sub>4</sub>-T<sub>6</sub> of OF cycle whereas  $\overline{\text{IO}/\overline{\text{M}}}$ , S<sub>0</sub>, and S<sub>1</sub> are stable.
2. Test conditions: t<sub>cyC</sub> = 320ns (8085A)/200ns (8085A-2); C<sub>L</sub> = 150pF.
3. For all output timing where C<sub>L</sub> = 150pF use the following correction factors:  
25pF ≤ C<sub>L</sub> < 150pF: -0.10 ns/pF  
150pF < C<sub>L</sub> ≤ 300pF: +0.30 ns/pF
4. Output timings are measured with purely capacitive load.
5. All timings are measured at output voltage V<sub>L</sub> = 0.8V, V<sub>H</sub> = 2.0V, and 1.5V with 20ns rise and fall time on inputs.
6. To calculate timing specifications at other values of t<sub>cyC</sub> use Table 7.
7. Data hold time is guaranteed under all loading conditions.

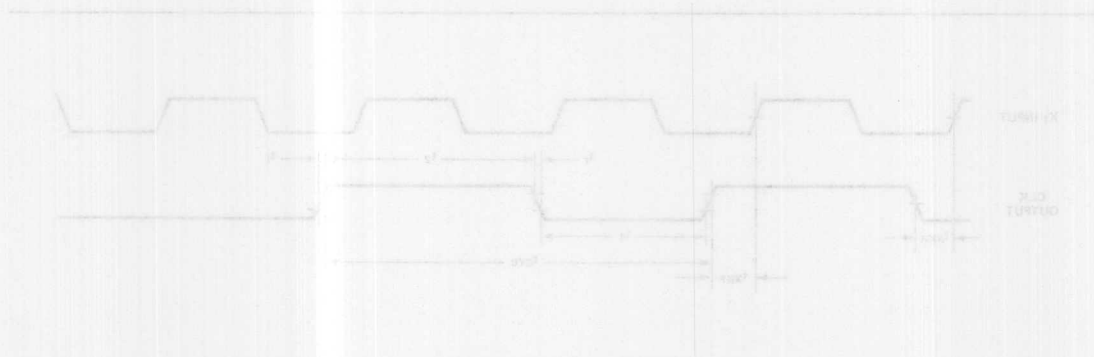


Figure 10. Clock Timing Waveform

TABLE 7. BUS TIMING SPECIFICATION AS A  $T_{CYC}$  DEPENDENT

8085A				8085A-2 (Preliminary)			
$t_{AL}$	—	$(1/2) T - 45$	MIN	$t_{AL}$	—	$(1/2) T - 50$	MIN
$t_{LA}$	—	$(1/2) T - 60$	MIN	$t_{LA}$	—	$(1/2) T - 50$	MIN
$t_{LL}$	—	$(1/2) T - 20$	MIN	$t_{LL}$	—	$(1/2) T - 20$	MIN
$t_{LCK}$	—	$(1/2) T - 60$	MIN	$t_{LCK}$	—	$(1/2) T - 50$	MIN
$t_{LC}$	—	$(1/2) T - 30$	MIN	$t_{LC}$	—	$(1/2) T - 40$	MIN
$t_{AD}$	—	$(5/2 + N) T - 225$	MAX	$t_{AD}$	—	$(5/2 + N) T - 150$	MAX
$t_{RD}$	—	$(3/2 + N) T - 180$	MAX	$t_{RD}$	—	$(3/2 + N) T - 150$	MAX
$t_{RAE}$	—	$(1/2) T - 10$	MIN	$t_{RAE}$	—	$(1/2) T - 10$	MIN
$t_{CA}$	—	$(1/2) T - 40$	MIN	$t_{CA}$	—	$(1/2) T - 40$	MIN
$t_{DW}$	—	$(3/2 + N) T - 60$	MIN	$t_{DW}$	—	$(3/2 + N) T - 70$	MIN
$t_{WD}$	—	$(1/2) T - 60$	MIN	$t_{WD}$	—	$(1/2) T - 40$	MIN
$t_{CC}$	—	$(3/2 + N) T - 80$	MIN	$t_{CC}$	—	$(3/2 + N) T - 70$	MIN
$t_{CL}$	—	$(1/2) T - 110$	MIN	$t_{CL}$	—	$(1/2) T - 75$	MIN
$t_{ARY}$	—	$(3/2) T - 260$	MAX	$t_{ARY}$	—	$(3/2) T - 200$	MAX
$t_{HACK}$	—	$(1/2) T - 50$	MIN	$t_{HACK}$	—	$(1/2) T - 60$	MIN
$t_{HABF}$	—	$(1/2) T + 50$	MAX	$t_{HABF}$	—	$(1/2) T + 50$	MAX
$t_{HABE}$	—	$(1/2) T + 50$	MAX	$t_{HABE}$	—	$(1/2) T + 50$	MAX
$t_{AC}$	—	$(2/2) T - 50$	MIN	$t_{AC}$	—	$(2/2) T - 85$	MIN
$t_1$	—	$(1/2) T - 80$	MIN	$t_1$	—	$(1/2) T - 60$	MIN
$t_2$	—	$(1/2) T - 40$	MIN	$t_2$	—	$(1/2) T - 30$	MIN
$t_{RV}$	—	$(3/2) T - 80$	MIN	$t_{RV}$	—	$(3/2) T - 80$	MIN
$t_{LDR}$	—	$(4/2) T - 180$	MAX	$t_{LDR}$	—	$(4/2) T - 130$	MAX

NOTE: N is equal to the total WAIT states.  
 $T = t_{CYC}$ .

NOTE: N is equal to the total WAIT states.  
 $T = t_{CYC}$ .

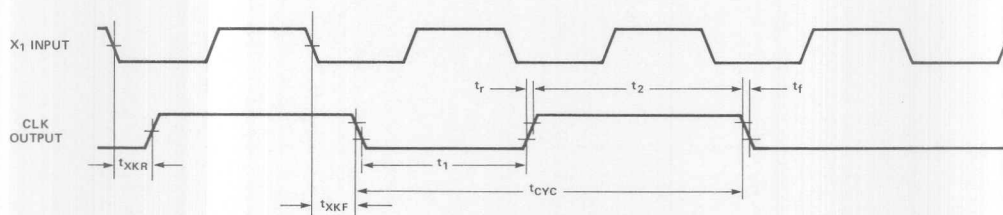
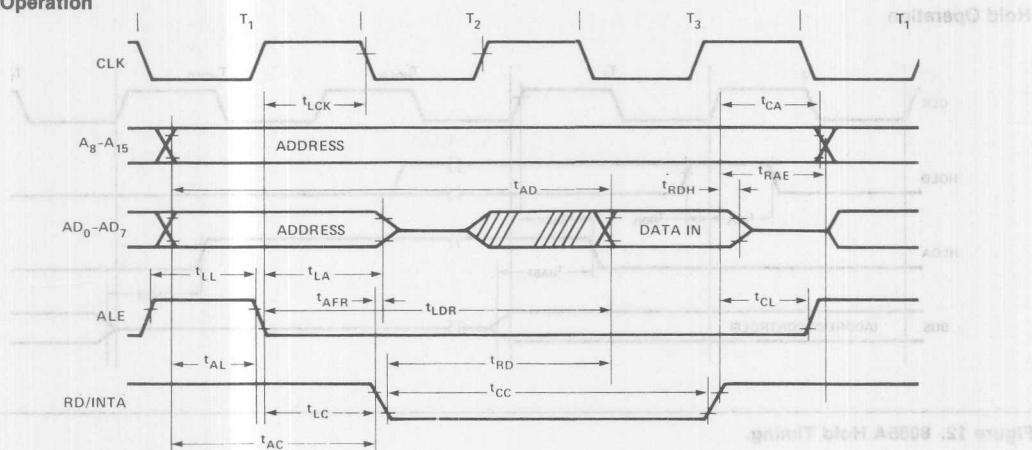
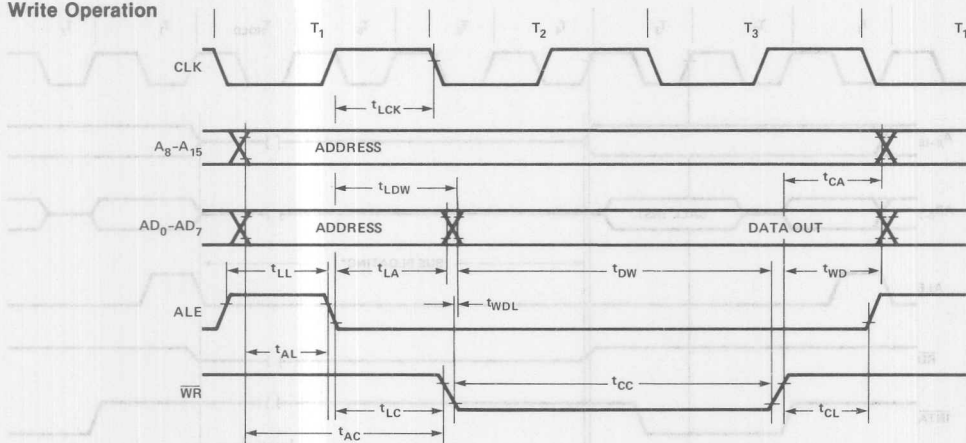


Figure 10. Clock Timing Waveform

## Read Operation



## Write Operation



Read operation with Wait Cycle (Typical) — same READY timing applies to WRITE operation.

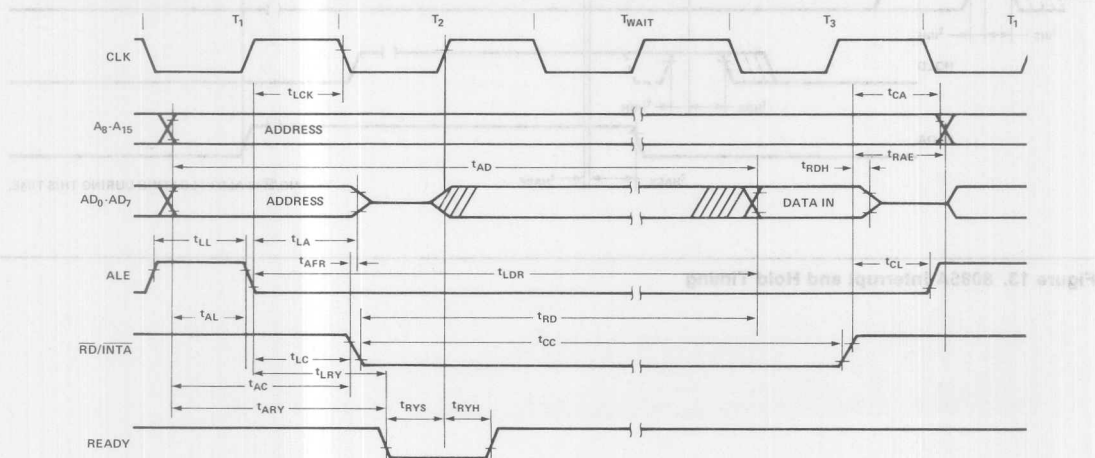


Figure 11. 8085A Bus Timing, With and Without Wait

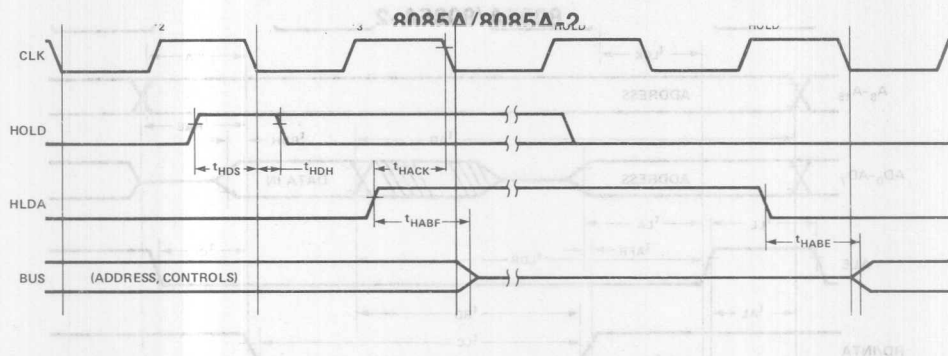


Figure 12. 8085A Hold Timing.

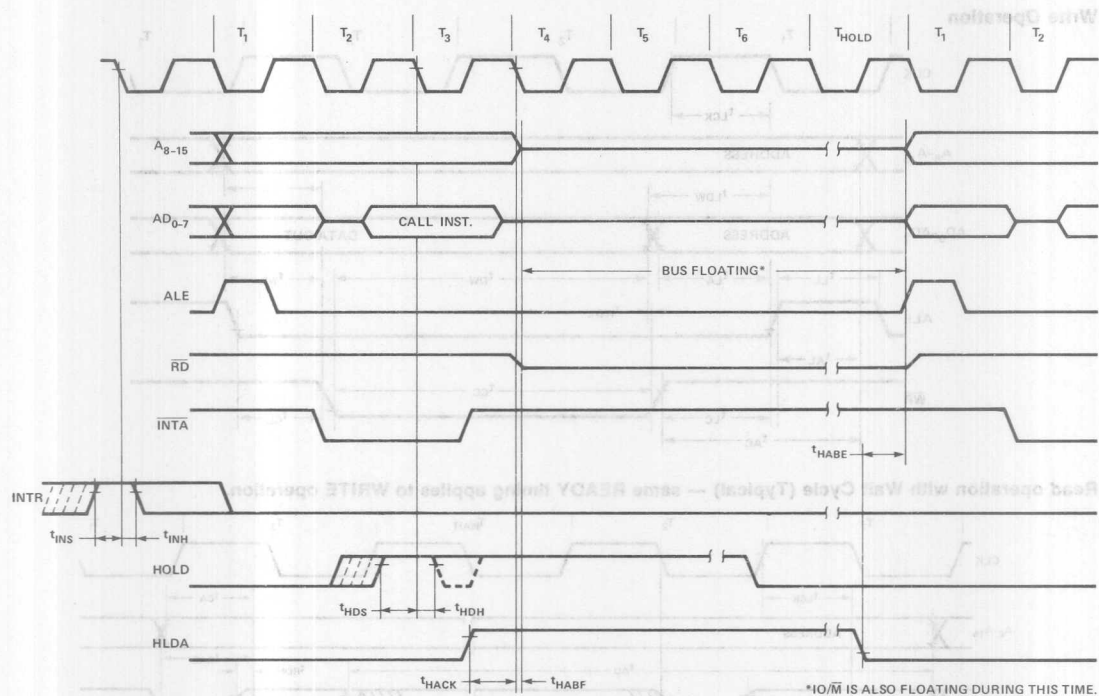


Figure 13. 8085A Interrupt and Hold Timing

TABLE 8. INSTRUCTION SET SUMMARY

Mnemonic	Description	Instruction Code[1]							Clock[2]	
		D <sub>7</sub>	D <sub>6</sub>	D <sub>5</sub>	D <sub>4</sub>	D <sub>3</sub>	D <sub>2</sub>	D <sub>1</sub>	D <sub>0</sub>	Cycles
MOVE, LOAD, AND STORE										
MOV r1 r2	Move register to register	0	1	D	D	D	S	S	S	4
MOV M, r	Move register to memory	0	1	1	1	0	S	S	S	7
MOV r, M	Move memory to register	0	1	D	D	D	1	1	0	7
MVI r	Move immediate register	0	0	D	D	D	1	1	0	7
MVI M	Move immediate memory	0	0	1	1	0	1	1	0	10
LXI B	Load immediate register Pair B & C	0	0	0	0	0	0	0	1	10
LXI D	Load immediate register Pair D & E	0	0	0	1	0	0	0	1	10
LXI H	Load immediate register Pair H & L	0	0	1	0	0	0	0	1	10
LXI SP	Load immediate stack pointer	0	0	1	1	0	0	0	1	10
STAX B	Store A indirect	0	0	0	0	0	0	1	0	7
STAX D	Store A indirect	0	0	0	1	0	0	1	0	7
LDAX B	Load A indirect	0	0	0	0	1	0	1	0	7
LDAX D	Load A indirect	0	0	0	1	1	0	1	0	7
STA	Store A direct	0	0	1	1	0	0	1	0	13
LDA	Load A direct	0	0	1	1	1	0	1	0	13
SHLD	Store H & L direct	0	0	1	0	0	0	0	1	16
LHLD	Load H & L direct	0	0	1	0	1	0	1	0	16
XCHG	Exchange D & E, H & L Registers	1	1	1	0	1	0	1	1	4
STACK OPS										
PUSH B	Push register Pair B & C on stack	1	1	0	0	0	1	0	1	12
PUSH D	Push register Pair D & E on stack	1	1	0	1	0	1	0	1	12
PUSH H	Push register Pair H & L on stack	1	1	1	0	0	1	0	1	12
PUSH PSW	Push A and Flags on stack	1	1	1	1	0	1	0	1	12
POP B	Pop register Pair B & C off stack	1	1	0	0	0	0	0	1	10
POP D	Pop register Pair D & E off stack	1	1	0	1	0	0	0	1	10
POP H	Pop register Pair H & L off stack	1	1	1	0	0	0	0	1	10
POP PSW	Pop A and Flags off stack	1	1	1	1	0	0	0	1	10
XTHL	Exchange top of stack, H & L	1	1	1	0	0	0	1	1	16
SPHL	H & L to stack pointer	1	1	1	1	1	0	0	1	6
JUMP										
JMP	Jump unconditional	1	1	0	0	0	0	1	1	10
JC	Jump on carry	1	1	0	1	1	0	1	0	7/10
JNC	Jump on no carry	1	1	0	1	0	0	1	0	7/10
JZ	Jump on zero	1	1	0	0	1	0	1	0	7/10
JNZ	Jump on no zero	1	1	0	0	0	0	1	0	7/10
JP	Jump on positive	1	1	1	1	0	0	1	0	7/10
JM	Jump on minus	1	1	1	1	1	0	1	0	7/10
JPE	Jump on parity even	1	1	1	0	1	0	1	0	7/10
JPO	Jump on parity odd	1	1	1	0	0	0	1	0	7/10
PCHL	H & L to program counter	1	1	1	0	1	0	0	1	6
CALL										
CALL	Call unconditional	1	1	0	0	1	1	0	1	18
CC	Call on carry	1	1	0	1	1	1	0	0	9/18
CNC	Call on no carry	1	1	0	1	0	1	0	0	9/18
CZ	Call on zero	1	1	0	0	1	1	0	0	9/18
CNZ	Call on no zero	1	1	0	0	0	1	0	0	9/18
CP	Call on positive	1	1	1	1	0	1	0	0	9/18
CM	Call on minus	1	1	1	1	1	1	0	0	9/18
Instruction Code[1]										
Clock[2]										
Cycles										
RETURN										
RET	Return	1	1	0	0	1	0	0	1	10
RC	Return on carry	1	1	0	1	1	0	0	0	6/12
RNC	Return on no carry	1	1	0	1	0	0	0	0	6/12
RZ	Return on zero	1	1	0	0	1	0	0	0	6/12
RNZ	Return on no zero	1	1	0	0	0	0	0	0	6/12
RP	Return on positive	1	1	1	1	0	0	0	0	6/12
RM	Return on minus	1	1	1	1	1	0	0	0	6/12
RPE	Return on parity even	1	1	1	0	1	0	0	0	6/12
RPO	Return on parity odd	1	1	1	0	0	0	0	0	6/12
RESTART										
RST	Restart	1	1	A	A	A	1	1	1	12
INPUT/OUTPUT										
IN	Input	1	1	0	1	1	0	1	1	10
OUT	Output	1	1	0	1	0	0	1	1	10
INCREMENT AND DECREMENT										
INR r	Increment register	0	0	D	D	D	1	0	0	4
DCR r	Decrement register	0	0	D	D	D	1	0	1	4
INR M	Increment memory	0	0	1	1	0	1	0	0	10
DCR M	Decrement memory	0	0	1	1	0	1	0	1	10
INX B	Increment B & C registers	0	0	0	0	0	0	1	1	6
INX D	Increment D & E registers	0	0	0	1	0	0	1	1	6
INX H	Increment H & L registers	0	0	1	0	0	0	1	1	6
INX SP	Increment stack pointer	0	0	1	1	0	0	1	1	6
DCX B	Decrement B & C	0	0	0	0	1	0	1	1	6
DCX D	Decrement D & E	0	0	0	1	1	0	1	1	6
DCX H	Decrement H & L	0	0	1	0	1	0	1	1	6
DCX SP	Decrement stack pointer	0	0	1	1	1	0	1	1	6
ADD										
ADD r	Add register to A	1	0	0	0	0	S	S	S	4
ADC r	Add register to A with carry	1	0	0	0	1	S	S	S	4
ADD M	Add memory to A	1	0	0	0	0	1	1	0	7
ADC M	Add memory to A with carry	1	0	0	0	1	1	1	0	7
ADI	Add immediate to A	1	1	0	0	0	1	1	0	7
ACI	Add immediate to A with carry	1	1	0	0	1	1	1	0	7
DAD B	Add B & C to H & L	0	0	0	0	1	0	0	1	10
DAD D	Add D & E to H & L	0	0	0	1	1	0	0	1	10
DAD H	Add H & L to H & L	0	0	1	0	1	0	0	1	10
DAD SP	Add stack pointer to H & L	0	0	1	1	1	0	0	1	10
SUBTRACT										
SUB r	Subtract register from A	1	0	0	1	0	S	S	S	4
SBB r	Subtract register from A with borrow	1	0	0	1	1	S	S	S	4
SUB M	Subtract memory from A	1	0	0	1	0	1	1	0	7
SBB M	Subtract memory from A with borrow	1	0	0	1	1	1	1	0	7
SUI	Subtract immediate from A	1	1	0	1	0	1	1	0	7
SBI	Subtract immediate from A with borrow	1	1	0	1	1	1	1	0	7
LOGICAL										
ANA r	And register with A	1	0	1	0	0	S	S	S	4

TABLE 8. INSTRUCTION SET SUMMARY (Continued)

Mnemonic	Description	Instruction Code[1]								Clock[2] Cycles
		D <sub>7</sub>	D <sub>6</sub>	D <sub>5</sub>	D <sub>4</sub>	D <sub>3</sub>	D <sub>2</sub>	D <sub>1</sub>	D <sub>0</sub>	
XRA r	Exclusive Or register with A	1	0	1	0	1	S	S	S	4
ORA r	Or register with A	1	0	1	1	0	S	S	S	4
CMP r	Compare register with A	1	0	1	1	1	S	S	S	4
ANA M	And memory with A	1	0	1	0	0	1	1	0	7
XRA M	Exclusive Or memory with A	1	0	1	0	1	1	1	0	7
ORA M	Or memory with A	1	0	1	1	0	1	1	0	7
CMP M	Compare memory with A	1	0	1	1	1	1	1	0	7
ANI	And immediate with A	1	1	1	0	0	1	1	0	7
XRI	Exclusive Or immediate with A	1	1	1	0	1	1	1	0	7
ORI	Or immediate with A	1	1	1	1	0	1	1	0	7
CPI	Compare immediate with A	1	1	1	1	1	1	1	0	7
<b>ROTATE</b>										
RLC	Rotate A left	0	0	0	0	0	1	1	1	4
RRC	Rotate A right	0	0	0	0	1	1	1	1	4
<b>SPECIALS</b>										
CMA	Complement A	0	0	1	0	1	1	1	1	4
STC	Set carry	0	0	1	1	0	1	1	1	4
CMC	Complement carry	0	0	1	1	1	1	1	1	4
DAA	Decimal adjust A	0	0	1	0	0	1	1	1	4
<b>CONTROL</b>										
EI	Enable Interrupts	1	1	1	1	1	0	1	1	4
DI	Disable Interrupt	1	1	1	1	0	0	1	1	4
NOP	No-operation	0	0	0	0	0	0	0	0	4
HLT	Halt	0	1	1	1	0	1	1	0	5
<b>NEW 8085A INSTRUCTIONS</b>										
RIM	Read Interrupt Mask	0	0	1	0	0	0	0	0	4
SIM	Set Interrupt Mask	0	0	1	1	0	0	0	0	4

NOTES: 1. DDD or SSS: B-000, C-001, D-010, E-011, H-100, L-101, Memory-110, A-111.

2. Two possible cycle times. (6/12) indicate instruction cycles dependent on condition flags.

\*All mnemonics copyright ©Intel Corporation 1977.

# 8155/8156/8155-2/8156-2

## 2048 BIT STATIC MOS RAM WITH I/O PORTS AND TIMER

8085A	8085A-2	Compatible CPU	Chip Enable
8155	8155-2	ACTIVE LOW	
8156	8156-2	ACTIVE HIGH	

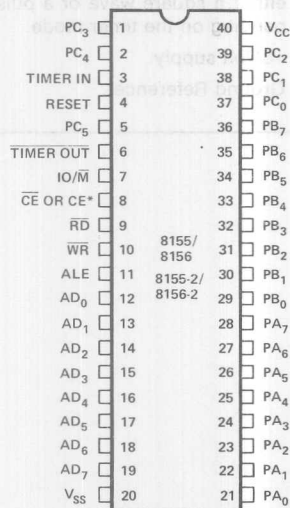
- 256 Word x 8 Bits
- Single +5V Power Supply
- Completely Static Operation
- Internal Address Latch
- 2 Programmable 8 Bit I/O Ports
- 1 Programmable 6-Bit I/O Port
- Programmable 14-Bit Binary Counter/Timer
- Multiplexed Address and Data Bus
- 40 Pin DIP

The 8155 and 8156 are RAM and I/O chips to be used in the MCS-85™ microcomputer system. The RAM portion is designed with 2048 static cells organized as 256 x 8. They have a maximum access time of 400 ns to permit use with no wait states in 8085A CPU. The 8155-2 and 8156-2 have maximum access times of 330 ns for use with the 8085A-2.

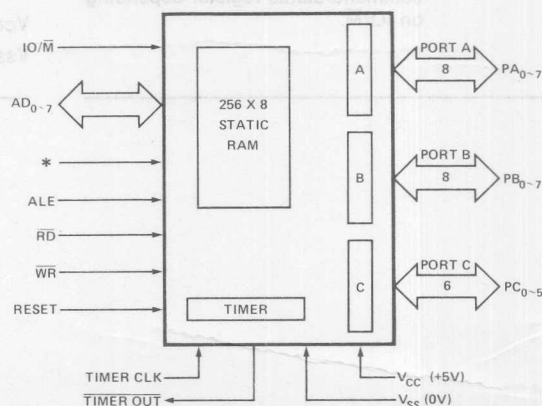
The I/O portion consists of three general purpose I/O ports. One of the three ports can be programmed to be status pins, thus allowing the other two ports to operate in handshake mode.

A 14-bit programmable counter/timer is also included on chip to provide either a square wave or terminal count pulse for the CPU system depending on timer mode.

### PIN CONFIGURATION



### BLOCK DIAGRAM



\*: 8155/8155-2 =  $\overline{CE}$ , 8156/8156-2 = CE

## 8155/8156 PIN FUNCTIONS

Symbol	Function	Symbol	Function
RESET (input)	Pulse provided by the 8085A to initialize the system (connect to 8085A RESET OUT). Input high on this line resets the chip and initializes the three I/O ports to input mode. The width of RESET pulse should typically be two 8085A clock cycle times.	ALE (input)	Address Latch Enable: This control signal latches both the address on the AD <sub>0-7</sub> lines and the state of the Chip Enable and IO/ $\overline{M}$ into the chip at the falling edge of ALE.
AD <sub>0-7</sub> (input)	3-state Address/Data lines that interface with the CPU lower 8-bit Address/Data Bus. The 8-bit address is latched into the address latch inside the 8155/56 on the falling edge of ALE. The address can be either for the memory section or the I/O section depending on the IO/ $\overline{M}$ input. The 8-bit data is either written into the chip or read from the chip, depending on the $\overline{WR}$ or $\overline{RD}$ input signal.	IO/ $\overline{M}$ (input)	Selects memory if low and I/O and command/status registers if high.
CE or $\overline{CE}$ (input)	Chip Enable: On the 8155, this pin is $\overline{CE}$ and is ACTIVE LOW. On the 8156, this pin is CE and is ACTIVE HIGH.	PA <sub>0-7</sub> (8) (input/output)	These 8 pins are general purpose I/O pins. The in/out direction is selected by programming the command register.
$\overline{RD}$ (input)	Read control: Input low on this line with the Chip Enable active enables and AD <sub>0-7</sub> buffers. If IO/ $\overline{M}$ pin is low, the RAM content will be read out to the AD bus. Otherwise the content of the selected I/O port or command/status registers will be read to the AD bus.	PB <sub>0-7</sub> (8) (input/output)	These 8 pins are general purpose I/O pins. The in/out direction is selected by programming the command register.
$\overline{WR}$ (input)	Write control: Input low on this line with the Chip Enable active causes the data on the Address/Data bus to be written to the RAM or I/O ports and command/status register depending on IO/ $\overline{M}$ .	PC <sub>0-5</sub> (6) (input/output)	These 6 pins can function as either input port, output port, or as control signals for PA and PB. Programming is done through the command register. When PC <sub>0-5</sub> are used as control signals, they will provide the following: PC <sub>0</sub> — A INTR (Port A Interrupt) PC <sub>1</sub> — ABF (Port A Buffer Full) PC <sub>2</sub> — A STB (Port A Strobe) PC <sub>3</sub> — B INTR (Port B Interrupt) PC <sub>4</sub> — B BF (Port B Buffer Full) PC <sub>5</sub> — B STB (Port B Strobe)
		TIMER IN (input)	Input to the counter-timer.
		$\overline{\text{TIMER OUT}}$ (output)	Timer output. This output can be either a square wave or a pulse depending on the timer mode.
		V <sub>CC</sub>	+5 volt supply.
		V <sub>SS</sub>	Ground Reference.

## DESCRIPTION

The 8155/8156 contains the following:

- 2k Bit Static RAM organized as 256 x 8
- Two 8-bit I/O ports (PA & PB) and one 6-bit I/O port (PC)
- 14-bit timer-counter

The  $\text{IO}/\overline{\text{M}}$  (IO/Memory Select) pin selects either the five registers (Command, Status, PA0-7, PB0-7, PC0-5) or the memory (RAM) portion. (See Figure 1.)

The 8-bit address on the Address/Data lines, Chip Enable input CE or  $\overline{\text{CE}}$ , and  $\text{IO}/\overline{\text{M}}$  are all latched on-chip at the falling edge of ALE. (See Figure 2.)

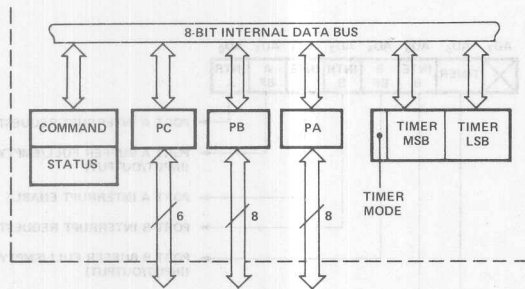
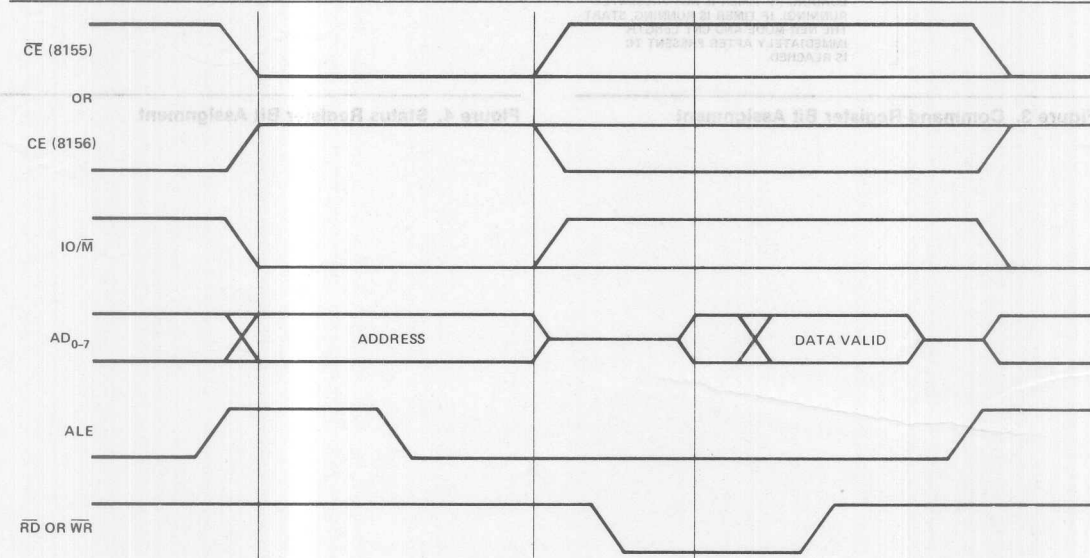


Figure 1. 8155/8156 Internal Registers



NOTE: FOR DETAILED TIMING INFORMATION, SEE FIGURE 12 AND A.C. CHARACTERISTICS.

Figure 2. 8155/8156 On-Board Memory Read/Write Cycle

The command register consists of eight latches. Four bits (0-3) define the mode of the ports, two bits (4-5) enable or disable the interrupt from port C when it acts as control port, and the last two bits (6-7) are for the timer.

The command register contents can be altered at any time by using the I/O address XXXXX000 during a WRITE operation with the Chip Enable active and  $IO/\overline{M} = 1$ . The meaning of each bit of the command byte is defined in Figure 3. The contents of the command register may never be read.

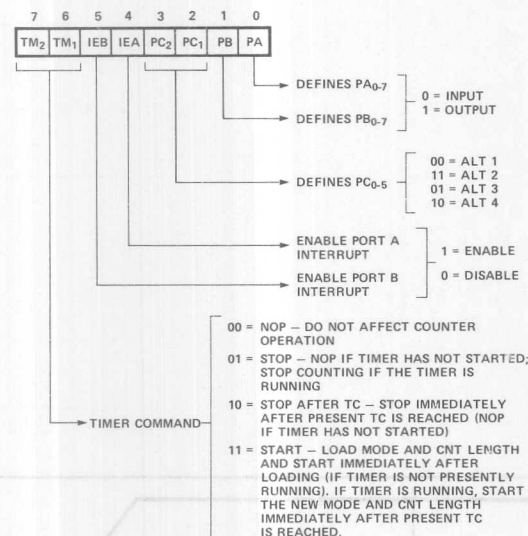


Figure 3. Command Register Bit Assignment

The status register consists of seven latches, one for each bit; six (0-5) for the status of the ports and one (6) for the status of the timer.

The status of the timer and the I/O section can be polled by reading the Status Register (Address XXXXX000). Status word format is shown in Figure 4. Note that you may never write to the status register since the command register shares the same I/O address and the command register is selected when a write to that address is issued.

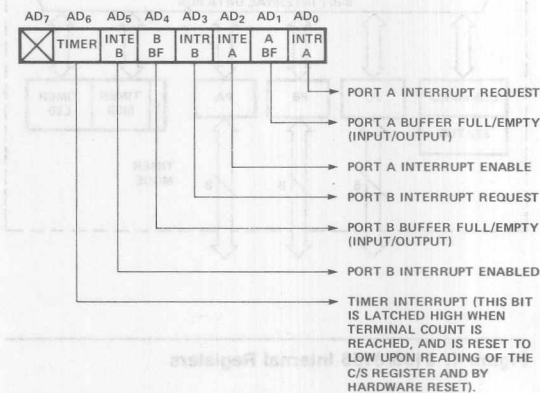


Figure 4. Status Register Bit Assignment

## INPUT/OUTPUT SECTION

The I/O section of the 8155/8156 consists of five registers:  
(See Figure 5.)

- **Command/Status Register (C/S)** — Both registers are assigned the address XXXXX000. The C/S address serves the dual purpose.

When the C/S registers are selected during WRITE operation, a command is written into the command register. The contents of this register are *not* accessible through the pins.

When the C/S (XXXXX000) is selected during a READ operation, the status information of the I/O ports and the timer becomes available on the AD<sub>0-7</sub> lines.

- **PA Register** — This register can be programmed to be either input or output ports depending on the status of the contents of the C/S Register. Also depending on the command, this port can operate in either the basic mode or the strobed mode (See timing diagram). The I/O pins assigned in relation to this register are PA<sub>0-7</sub>. The address of this register is XXXXX001.
- **PB Register** — This register functions the same as PA Register. The I/O pins assigned are PB<sub>0-7</sub>. The address of this register is XXXXX010.
- **PC Register** — This register has the address XXXXX011 and contains only 6 bits. The 6 bits can be programmed to be either input ports, output ports or as control signals for PA and PB by properly programming the AD<sub>2</sub> and AD<sub>3</sub> bits of the C/S register.

When PC<sub>0-5</sub> is used as a control port, 3 bits are assigned for Port A and 3 for Port B. The first bit is an interrupt that the 8155 sends out. The second is an output signal indicating whether the buffer is full or empty, and the third is an input pin to accept a strobe for the strobed input mode. (See Table 1.)

When the 'C' port is programmed to either ALT3 or ALT4, the control signals for PA and PB are initialized as follows:

CONTROL	INPUT MODE	OUTPUT MODE
BF	Low	Low
INTR	Low	High
STB	Input Control	Input Control

I/O ADDRESS†							SELECTION	
A7	A6	A5	A4	A3	A2	A1	A0	
X	X	X	X	X	0	0	0	Interval Command/Status Register
X	X	X	X	X	0	0	1	General Purpose I/O Port A
X	X	X	X	X	0	1	0	General Purpose I/O Port B
X	X	X	X	X	0	1	1	Port C — General Purpose I/O or Control
X	X	X	X	X	1	0	0	Low-Order 8 bits of Timer Count
X	X	X	X	X	1	0	1	High 6 bits of Timer Count and 2 bits of Timer Mode

X: Don't Care.

†: I/O Address must be qualified by CE = 1 (8156) or  $\overline{\text{CE}} = 0$  (8155) and IO/ $\overline{\text{M}} = 1$  in order to select the appropriate register.

Figure 5. I/O port and Timer Addressing Scheme

Figure 6 shows how I/O PORTS A and B are structured within the 8155 and 8156:

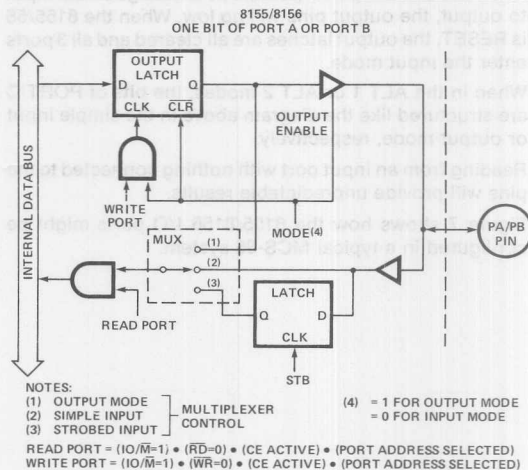


Figure 6. 8155/8156 Port Functions

TABLE 1. TABLE OF PORT CONTROL ASSIGNMENT.

Pin	ALT 1	ALT 2	ALT 3	ALT 4
PC0	Input Port	Output Port	A INTR (Port A Interrupt)	A INTR (Port A Interrupt)
PC1	Input Port	Output Port	A BF (Port A Buffer Full)	A BF (Port A Buffer Full)
PC2	Input Port	Output Port	A STB (Port A Strobe)	A STB (Port A Strobe)
PC3	Input Port	Output Port	Output Port	B INTR (Port B Interrupt)
PC4	Input Port	Output Port	Output Port	B BF (Port B Buffer Full)
PC5	Input Port	Output Port	Output Port	B STB (Port B Strobe)

Note in the diagram that when the I/O ports are programmed to be output ports, the contents of the output ports can still be read by a READ operation when appropriately addressed.

The outputs of the 8155/8156 are "glitch-free" meaning that you can write a "1" to a bit position that was previously "1" and the level at the output pin will not change.

Note also that the output latch is cleared when the port enters the input mode. The output latch cannot be loaded by writing to the port if the port is in the input mode. The result is that each time a port mode is changed from input to output, the output pins will go low. When the 8155/56 is RESET, the output latches are all cleared and all 3 ports enter the input mode.

When in the ALT 1 or ALT 2 modes, the bits of PORT C are structured like the diagram above in the simple input or output mode, respectively.

Reading from an input port with nothing connected to the pins will provide unpredictable results.

Figure 7 shows how the 8155/8156 I/O ports might be configured in a typical MCS-85 system.

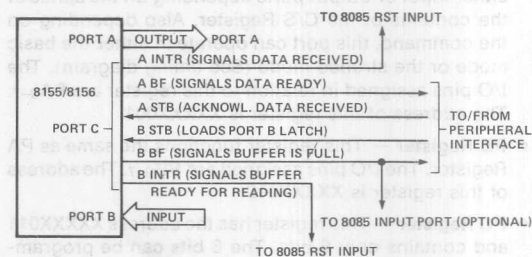


Figure 7. Example: Command Register = 00111001

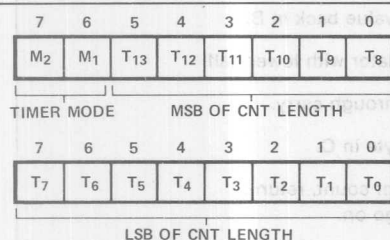
CONTROL	INPUT MODE	OUTPUT MODE
BF	Low	Low
INTR	Low	High
STB	Input Control	Input Control

## TIMER SECTION

The timer is a 14-bit down-counter that counts the TIMER IN pulses and provides either a square wave or pulse when terminal count (TC) is reached.

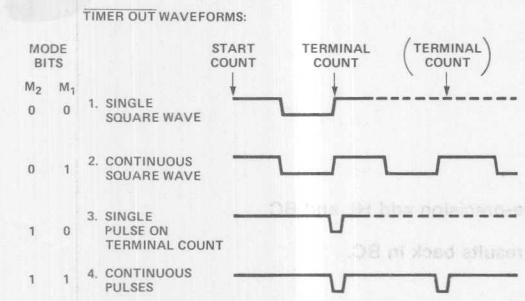
The timer has the I/O address XXXXX100 for the low order byte of the register and the I/O address XXXXX101 for the high order byte of the register. (See Figure 5).

To program the timer, the COUNT LENGTH REG is loaded first, one byte at a time, by selecting the timer addresses. Bits 0-13 of the high order count register will specify the length of the next count and bits 14-15 of the high order register will specify the timer output mode (see Figure 8). The value loaded into the count length register can have any value from 2H through 3FH in Bits 0-13.



**Figure 8. Timer Format**

There are four modes to choose from: M2 and M1 define the timer mode, as shown in Figure 9.



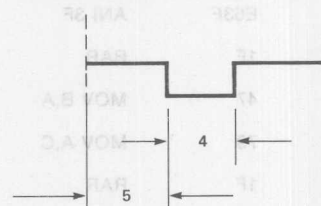
**Figure 9. Timer Modes**

Bits 6-7 (TM<sub>2</sub> and TM<sub>1</sub>) of command register contents are used to start and stop the counter. There are four commands to choose from:

TM <sub>2</sub>	TM <sub>1</sub>	
0	0	NOP — Do not affect counter operation.
0	1	STOP — NOP if timer has not started; stop counting if the timer is running.
1	0	STOP AFTER TC — Stop immediately after present TC is reached (NOP if timer has not started)
1	1	START — Load mode and CNT length and start immediately after loading (if timer is not presently running). If timer is running, start the new mode and CNT length immediately after present TC is reached.

Note that while the counter is counting, you may load a new count and mode into the count length registers. Before the new count and mode will be used by the counter, you must issue a START command to the counter. This applies even though you may only want to change the count and use the previous mode.

In case of an odd-numbered count, the first half-cycle of the squarewave output, which is high, is one count longer than the second (low) half-cycle, as shown in Figure 10.



NOTE: 5 AND 4 REFER TO THE NUMBER OF CLOCKS IN THAT TIME PERIOD

**Figure 10. Asymmetrical Square-Wave Output Resulting from Count of 9**

The counter in the 8155 is not initialized to any particular mode or count when hardware RESET occurs, but RESET does stop the counting. Therefore, counting cannot begin following RESET until a START command is issued via the C/S register.

Please note that the timer circuit on the 8155/8156 chip is designed to be a square-wave timer, not an event counter. To achieve this, it counts down by twos twice in completing one cycle. Thus, its registers do not contain values directly representing the number of TIMER IN pulses received. You cannot load an initial value of 1 into the count register and cause the timer to operate, as its terminal count value is 10 (binary) or 2 (decimal). (For the detection of single pulses, it is suggested that one of the hardware interrupt pins on the 8085A be used.) After the timer has started counting down, the values residing in the count registers can be used to calculate the actual number of TIMER IN pulses required to complete the timer cycle if desired. To obtain the remaining count, perform the following operations in order:

1. Stop the count
2. Read in the 16-bit value from the count length registers
3. Reset the upper two mode bits
4. Reset the carry and rotate right one position all 16 bits through carry
5. If carry is set, add 1/2 of the full original count (1/2 full count — 1 if full count is odd).

Note: If you started with an odd count and you read the count length register before the third count pulse occurs, you will not be able to discern whether one or two counts has occurred. Regardless of this, the 8155/56 always counts out the right number of pulses in generating the TIMER OUT waveforms.

## EXAMPLE PROGRAM

Following is an actual sequence of program steps that adjusts the 8155/56 count register contents to obtain the count, extracted from Intel® Application Note AP38. "Application Techniques for the Intel 8085A Bus." First store the value of the full original count in register HL of the 8085A. Then stop the count to avoid getting an incorrect count value. Then sample the timer-counter, storing the lower-order byte of the current count register in register C and the higher-order count byte in register B. Then, call the following 8080A/8085A subroutine:

ADJUST, 78	MOV A,B	;Load accumulator with upper half ; of count.
E63F	ANI 3F	;Reset upper 2 bits and clear carry.
1F	RAR	;Rotate right through carry.
47	MOV B,A	;Store shifted value back in B.
79	MOV A,C	;Load accumulator with lower half.
1F	RAR	;Rotate right through carry.
4F	MOV C,A	;Store lower byte in C.
D0	RNC	;If in 2nd half of count, return. ;If in 1st half, go on.
3F	CMC	;Clear carry.
7C	MOV A,H	;Divide full count by 2. (If HL ;is odd, disregard remainder.)
1F	RAR	
67	MOV H,A	
7D	MOV A,L	
1F	RAR	
6F	MOV L,A	
09	DAD B	;Double-precision add HL and BC.
44	MOV B,H	;Store results back in BC.
4D	MOV C,L	
C9	RET	;Return.

After executing the subroutine, BC will contain the remaining count in the current count cycle.

- 256 Bytes RAM
- 2K Bytes ROM
- 38 I/O Pins
- 1 Interval Timer
- 4 Interrupt Levels



Maximum Ratings" may cause permanent damage to the device. This is a stress rating only and functional operation of the device at these or any other conditions above those indicated in the operational sections of this specification is not implied. Exposure to absolute maximum rating conditions for extended periods may affect device reliability.

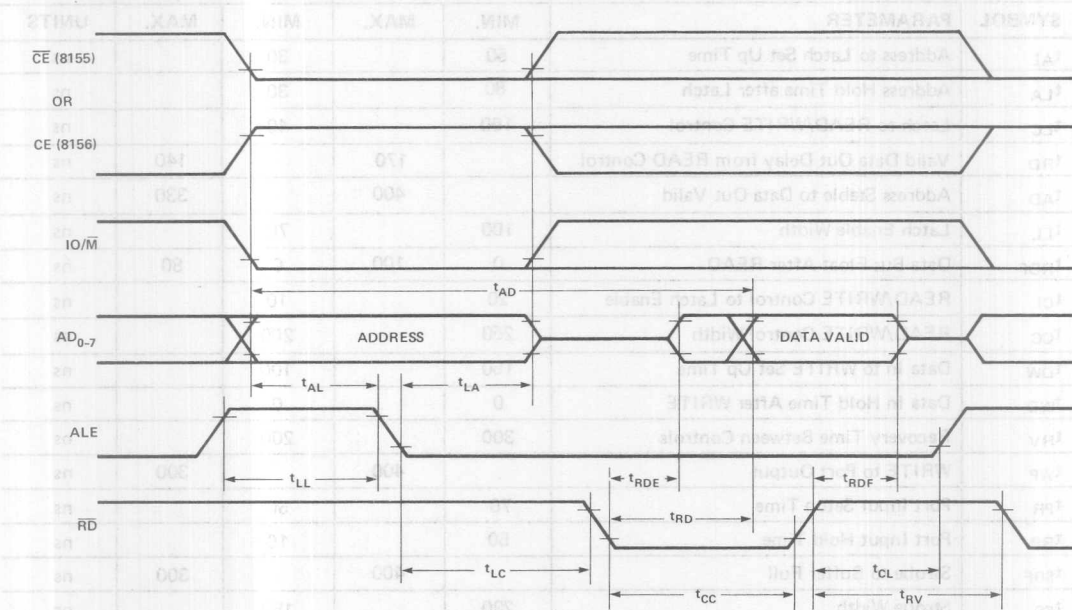
SYMBOL	PARAMETER	MIN.	MAX.	UNITS	TEST CONDITIONS
V <sub>IL</sub>	Input Low Voltage	-0.5	0.8	V	
V <sub>IH</sub>	Input High Voltage	2.0	V <sub>CC</sub> +0.5	V	
V <sub>OL</sub>	Output Low Voltage		0.45	V	I <sub>OL</sub> = 2mA
V <sub>OH</sub>	Output High Voltage	2.4		V	I <sub>OH</sub> = -400μA
I <sub>IL</sub>	Input Leakage		±10	μA	V <sub>IN</sub> = V <sub>CC</sub> to 0V
I <sub>LO</sub>	Output Leakage Current		±10	μA	0.45V ≤ V <sub>OUT</sub> ≤ V <sub>CC</sub>
I <sub>CC</sub>	V <sub>CC</sub> Supply Current		180	mA	
I <sub>IL</sub> (CE)	Chip Enable Leakage				
	8155		+100	μA	V <sub>IN</sub> = V <sub>CC</sub> to 0V
	8156		-100	μA	

**A.C. CHARACTERISTICS** ( $T_A = 0^\circ\text{C}$  to  $70^\circ\text{C}$ ;  $V_{CC} = 5V \pm 5\%$ )

SYMBOL	PARAMETER	8155/8156		8155-2/8156-2 (Preliminary)		UNITS
		MIN.	MAX.	MIN.	MAX.	
$t_{AL}$	Address to Latch Set Up Time	50		30		ns
$t_{LA}$	Address Hold Time after Latch	80		30		ns
$t_{LC}$	Latch to READ/WRITE Control	100		40		ns
$t_{RD}$	Valid Data Out Delay from READ Control		170		140	ns
$t_{AD}$	Address Stable to Data Out Valid		400		330	ns
$t_{LL}$	Latch Enable Width	100		70		ns
$t_{RDF}$	Data Bus Float After READ	0	100	0	80	ns
$t_{CL}$	READ/WRITE Control to Latch Enable	20		10		ns
$t_{CC}$	READ/WRITE Control Width	250		200		ns
$t_{DW}$	Data In to WRITE Set Up Time	150		100		ns
$t_{WD}$	Data In Hold Time After WRITE	0		0		ns
$t_{RV}$	Recovery Time Between Controls	300		200		ns
$t_{WP}$	WRITE to Port Output		400		300	ns
$t_{PR}$	Port Input Setup Time	70		50		ns
$t_{RP}$	Port Input Hold Time	50		10		ns
$t_{SBF}$	Strobe to Buffer Full		400		300	ns
$t_{SS}$	Strobe Width	200		150		ns
$t_{RBE}$	READ to Buffer Empty		400		300	ns
$t_{SI}$	Strobe to INTR On		400		300	ns
$t_{RDI}$	READ to INTR Off		400		300	ns
$t_{PSS}$	Port Setup Time to Strobe Strobe	50		0		ns
$t_{PHS}$	Port Hold Time After Strobe	120		100		ns
$t_{SBE}$	Strobe to Buffer Empty		400		300	ns
$t_{WBF}$	WRITE to Buffer Full		400		300	ns
$t_{WI}$	WRITE to INTR Off		400		300	ns
$t_{TL}$	TIMER-IN to $\overline{\text{TIMER-OUT}}$ Low		400		300	ns
$t_{TH}$	TIMER-IN to $\overline{\text{TIMER-OUT}}$ High		400		300	ns
$t_{RDE}$	Data Bus Enable from READ Control	10		10		ns
$t_1$	TIMER-IN Low Time	80		40		ns
$t_2$	TIMER-IN High Time	120		70		ns

## WAVEFORMS

## a. Read Cycle



## b. Write Cycle

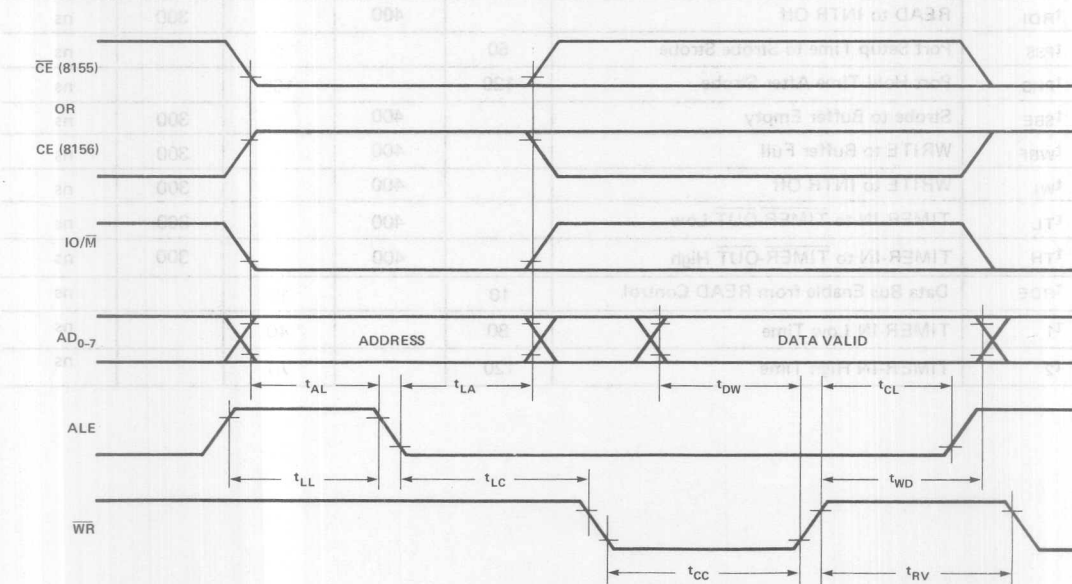
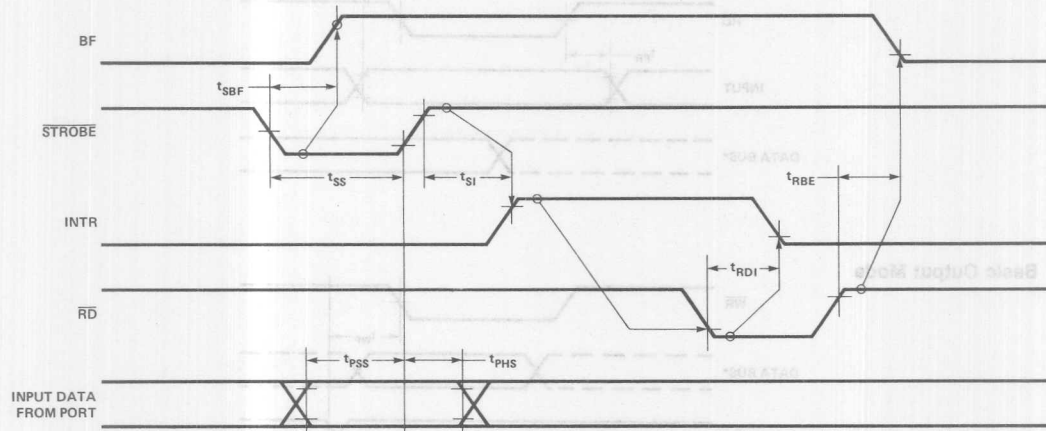


Figure 12. 8155/8156 Read/Write Timing Diagrams

## a. Strobed Input Mode



## b. Strobed Output Mode

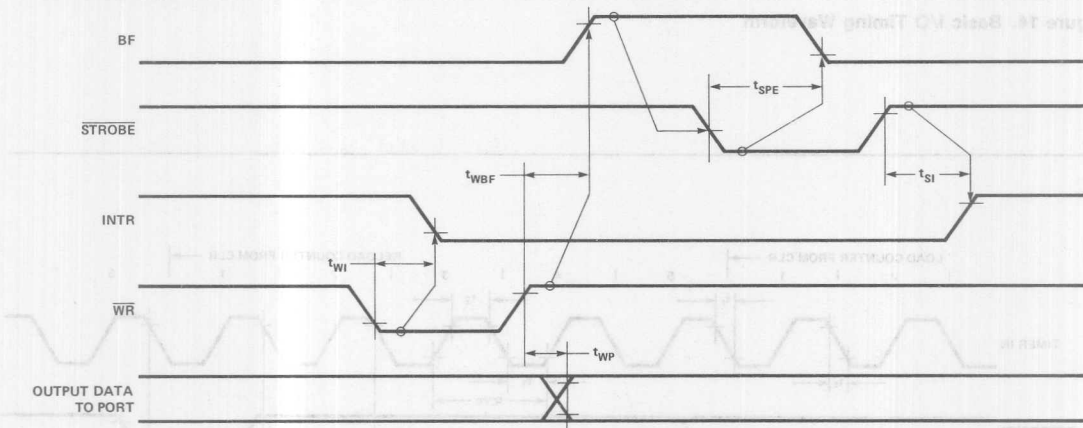
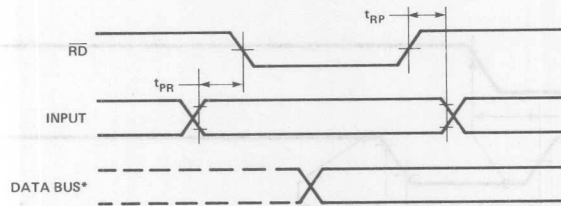
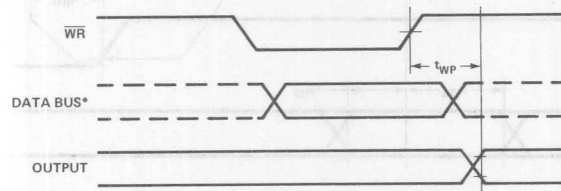


Figure 13. Strobed I/O Timing

## a. Basic Input Mode



## b. Basic Output Mode



\*DATA BUS TIMING IS SHOWN IN FIGURE 7.

Figure 14. Basic I/O Timing Waveform

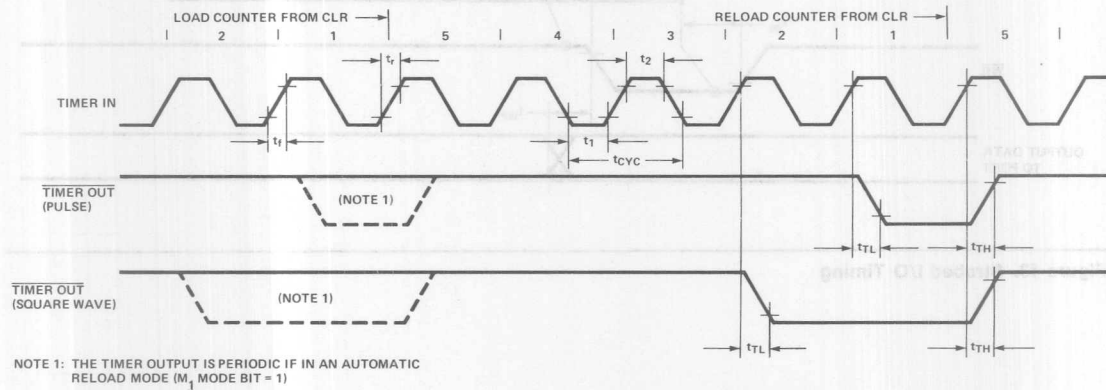


Figure 15. Timer Output Waveform Countdown from 5 to 1

# 8185\*/8185-2\*\* 1024 x 8-BIT STATIC RAM FOR MCS-85™

\*Compatible with 8085A

\*\*Compatible with 8085A-2

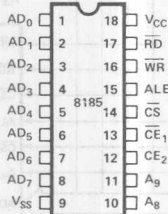
- Multiplexed Address and Data Bus
- Directly Compatible with 8085A Microprocessor
- Low Operating Power Dissipation
- Low Standby Power Dissipation
- Single +5V Supply
- High Density 18-Pin Package

The Intel® 8185 is an 8192-bit static random access memory (RAM) organized as 1024 words by 8-bits using N-channel Silicon-Gate MOS technology. The multiplexed address and data bus allows the 8185 to interface directly to the 8085A microprocessor to provide a maximum level of system integration.

The low standby power dissipation minimizes system power requirements when the 8185 is disabled.

The 8185-2 is a high-speed selected version of the 8185.

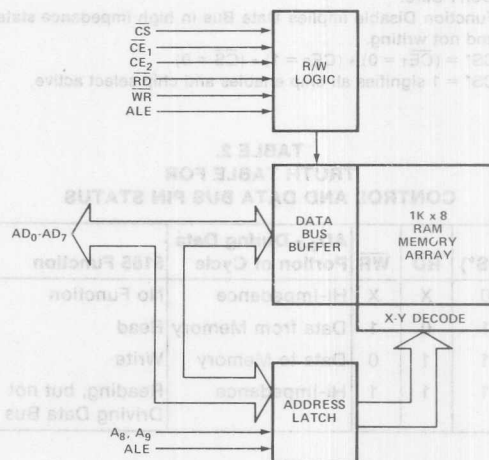
## PIN CONFIGURATION



## PIN NAMES

AD <sub>0</sub> -AD <sub>7</sub>	ADDRESS/DATA LINES
A <sub>8</sub> , A <sub>9</sub>	ADDRESS LINES
CS	CHIP SELECT
CE <sub>1</sub>	CHIP ENABLE (IO/ $\bar{M}$ )
CE <sub>2</sub>	CHIP ENABLE
ALE	ADDRESS LATCH ENABLE
RD	READ ENABLE
WR	WRITE ENABLE

## BLOCK DIAGRAM



At the beginning of an 8185 memory access cycle, the 8-bit address on AD<sub>0-7</sub>, A<sub>8</sub> and A<sub>9</sub>, and the status of  $\overline{CE}_1$  and  $\overline{CE}_2$  are all latched internally in the 8185 by the falling edge of ALE. If the latched status of both  $\overline{CE}_1$  and  $\overline{CE}_2$  are active, the 8185 powers itself up, but no action occurs until the  $\overline{CS}$  line goes low and the appropriate  $\overline{RD}$  or  $\overline{WR}$  control signal input is activated.

The  $\overline{CS}$  input is not latched by the 8185 in order to allow the maximum amount of time for address decoding in selecting the 8185 chip. Maximum power consumption savings will occur, however, only when  $\overline{CE}_1$  and  $\overline{CE}_2$  are activated selectively to power down the 8185 when it is not in use. A possible connection would be to wire the 8085A's IO/ $\overline{M}$  line to the 8185's  $\overline{CE}_1$  input, thereby keeping the 8185 powered down during I/O and interrupt cycles.

**TABLE 1.**  
**TRUTH TABLE FOR**  
**POWER DOWN AND FUNCTION ENABLE**

$\overline{CE}_1$	$\overline{CE}_2$	$\overline{CS}$	(CS*) <sup>[2]</sup>	8185 Status
1	X	X	0	Power Down and Function Disable <sup>[1]</sup>
X	0	X	0	Power Down and Function Disable <sup>[1]</sup>
0	1	1	0	Powered Up and Function Disable <sup>[1]</sup>
0	1	0	1	Powered Up and Enabled

Notes:

X: Don't Care.

1: Function Disable implies Data Bus in high impedance state and not writing.

2: CS\* = ( $\overline{CE}_1 = 0$ ) • ( $\overline{CE}_2 = 1$ ) • ( $\overline{CS} = 0$ )

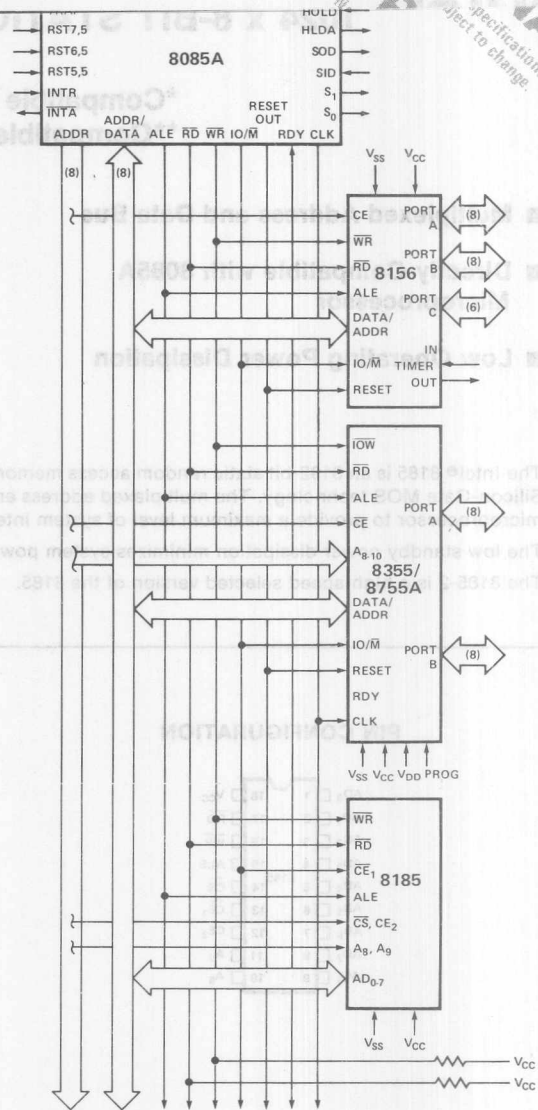
CS\* = 1 signifies all chip enables and chip select active

**TABLE 2.**  
**TRUTH TABLE FOR**  
**CONTROL AND DATA BUS PIN STATUS**

(CS*)	$\overline{RD}$	$\overline{WR}$	AD <sub>0-7</sub> During Data Portion of Cycle	8185 Function
0	X	X	Hi-Impedance	No Function
1	0	1	Data from Memory	Read
1	1	0	Data to Memory	Write
1	1	1	Hi-Impedance	Reading, but not Driving Data Bus

Note:

X: Don't Care.



**Figure 1. 8185 in an MCS-85 System.**

4 Chips:

2K Bytes ROM  
1.25K Bytes RAM  
38 I/O Lines  
1 Counter/Timer  
2 Serial I/O Lines  
5 Interrupt Inputs

**ABSOLUTE MAXIMUM RATINGS\***

Temperature Under Bias ..... 0°C to +70°C  
 Storage Temperature ..... -65°C to +150°C  
 Voltage on Any Pin  
   with Respect to Ground ..... -0.5V to +7V  
 Power Dissipation ..... 1.5W

**\*COMMENT:**

Stresses above those listed under "Absolute Maximum Ratings" may cause permanent damage to the device. This is a stress rating only and functional operation of the device at these or any other conditions above those indicated in the operational sections of this specification is not implied. Exposure to absolute maximum rating conditions for extended periods may affect device reliability.

**D.C. CHARACTERISTICS** ( $T_A = 0^\circ\text{C}$  to  $70^\circ\text{C}$ ;  $V_{CC} = 5V \pm 5\%$ )

Symbol	Parameter	Min.	Max.	Units	Test Conditions
$V_{IL}$	Input Low Voltage	-0.5	0.8	V	
$V_{IH}$	Input High Voltage	2.0	$V_{CC}+0.5$	V	
$V_{OL}$	Output Low Voltage		0.45	V	$I_{OL} = 2\text{mA}$
$V_{OH}$	Output High Voltage	2.4			$I_{OH} = 400\mu\text{A}$
$I_{IL}$	Input Leakage		$\pm 10$	$\mu\text{A}$	$V_{IN} = V_{CC}$ to 0V
$I_{LO}$	Output Leakage Current		$\pm 10$	$\mu\text{A}$	$0.45V \leq V_{OUT} \leq V_{CC}$
$I_{CC}$	$V_{CC}$ Supply Current Powered Up		100	mA	
	Powered Down		25	mA	

**A.C. CHARACTERISTICS** ( $T_A = 0^\circ\text{C}$  to  $70^\circ\text{C}$ ;  $V_{CC} = 5V \pm 5\%$ )

Symbol	Parameter <sup>[1]</sup>	8185 Preliminary		8185-2 Preliminary		Units
		Min.	Max.	Min.	Max.	
$t_{AL}$	Address to Latch Set Up Time	50		30		ns
$t_{LA}$	Address Hold Time After Latch	80		30		ns
$t_{LC}$	Latch to READ/WRITE Control	100		40		ns
$t_{RD}$	Valid Data Out Delay from READ Control	170		140		ns
$t_{LD}$	ALE to Data Out Valid	300		200		ns
$t_{LL}$	Latch Enable Width	100		70		ns
$t_{RDF}$	Data Bus Float After READ	0	100	0	80	ns
$t_{CL}$	READ/WRITE Control to Latch Enable	20		10		ns
$t_{CC}$	READ/WRITE Control Width	250		200		ns
$t_{DW}$	Data In to WRITE Set Up Time	150		150		ns
$t_{WD}$	Data In Hold Time After WRITE	20		20		ns
$t_{SC}$	Chip Select Set Up to Control Line	10		10		ns
$t_{CS}$	Chip Select Hold Time After Control	10		10		ns
$t_{ALCE}$	Chip Enable Set Up to ALE Falling	30		10		ns
$t_{LACE}$	Chip Enable Hold Time After ALE	50		30		ns

**Notes:**

1. All AC parameters are referenced at
  - a) 2.4V and .45V for inputs
  - b) 2.0V and .8V for outputs.

**PRELIMINARY**  
 Notice: This is not a final specification. Some parameters are subject to change.

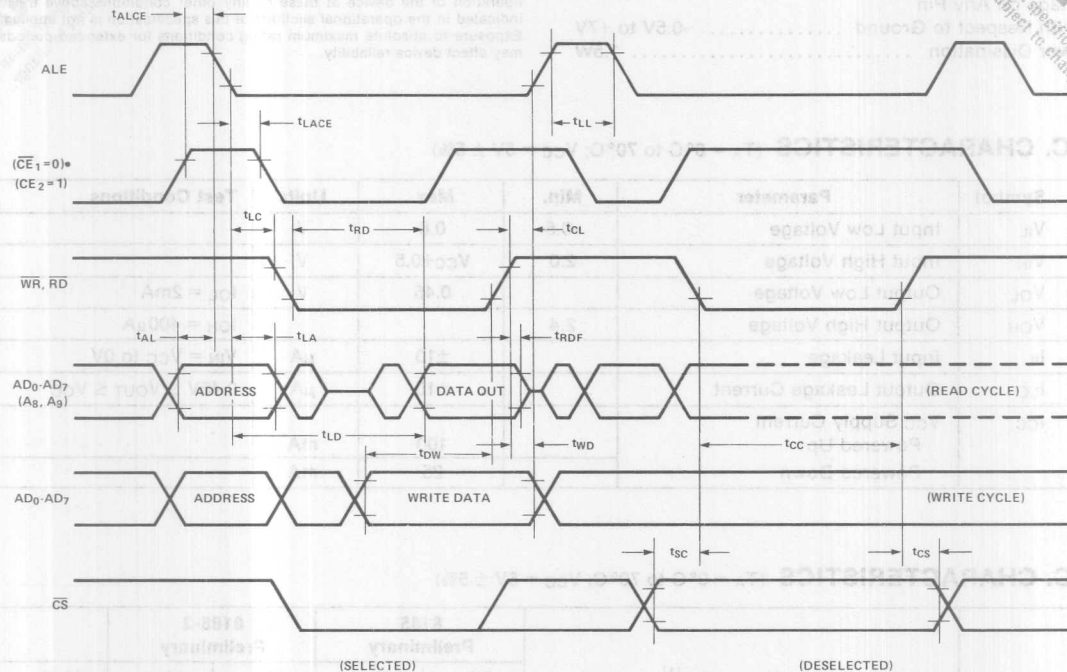


Figure 2. 8185 Timing.

Symbol	Parameter (1)	Min.	Max.	Unit	Notes
$t_{AL}$	Address to Latch Set Up Time	50	200	ns	
$t_{LA}$	Address Hold Time After Latch	50	100	ns	
$t_{LCE}$	Latch to READ/WRITE Control	100	150	ns	
$t_{RD}$	Valid Data Out Delay from READ Control	100	150	ns	
$t_{LD}$	ALE to Data Out Valid	100	150	ns	
$t_{LW}$	Latch Enable Width	100	150	ns	
$t_{RDf}$	Data Bus Float After READ	0	100	ns	
$t_{RDf}$	READ/WRITE Control to Latch Enable	50	100	ns	
$t_{RDf}$	READ/WRITE Control Width	250	300	ns	
$t_{WD}$	Data In to WRITE Set Up Time	150	200	ns	
$t_{WDf}$	Data In Hold Time After WRITE	50	100	ns	
$t_{sc}$	Chip Select Set Up to Control Line	10	20	ns	
$t_{cs}$	Chip Select Hold Time After Control	10	20	ns	
$t_{ALCE}$	Chip Enable Set Up to ALE Falling	50	100	ns	
$t_{LACE}$	Chip Enable Hold Time After ALE	50	100	ns	

Notes:  
 1. All AC parameters are referenced at 25°C and 10V for inputs and 5V for outputs.

# 8355\*/8355-2\*\*

## 16,384-BIT ROM WITH I/O

\*Directly Compatible with 8085A CPU

\*\*Directly Compatible with 8085A-2

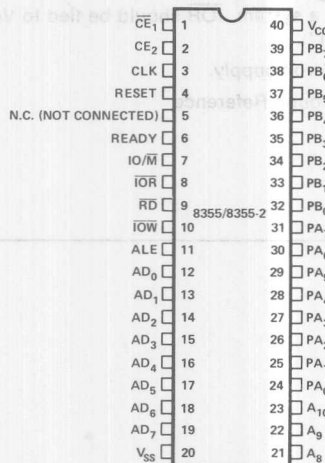
- 2048 Words x 8 Bits
- Single +5V Power Supply
- Internal Address Latch
- 2 General Purpose 8-Bit I/O Ports
- Each I/O Port Line Individually Programmable as Input or Output
- Multiplexed Address and Data Bus
- 40-Pin DIP

The Intel® 8355 is a ROM and I/O chip to be used in the MCS-85™ microcomputer system. The ROM portion is organized as 2048 words by 8 bits. It has a maximum access time of 400 ns to permit use with no wait states in the 8085A CPU.

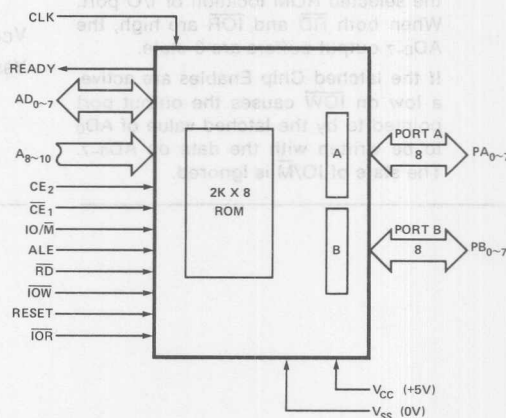
The I/O portion consists of 2 general purpose I/O ports. Each I/O port has 8 port lines, and each I/O port line is individually programmable as input or output.

The 8355-2 has a 300ns access time for compatibility with the 8085A-2 microprocessor.

### PIN CONFIGURATION



### BLOCK DIAGRAM



## 8355/8355-2

	enter address latched. The signals (AD, IO/ $\overline{M}$ , A <sub>8-10</sub> , CE, $\overline{CE}$ ) are latched in at the trailing edge of ALE.		
AD <sub>0-7</sub> (Input)	Bidirectional Address/Data bus. The lower 8-bits of the ROM or I/O address are applied to the bus lines when ALE is high.	READY (Output)	Ready is a 3-state output controlled by $\overline{CE}_1$ , CE <sub>2</sub> , ALE and CLK. READY is forced low when the Chip Enables are active during the time ALE is high, and remains low until the rising edge of the next CLK (see Figure 6).
	During an I/O cycle, Port A or B are selected based on the latched value of AD <sub>0</sub> . If $\overline{RD}$ or $\overline{IOR}$ is low when the latched chip enables are active, the output buffers present data on the bus.	PA <sub>0-7</sub> (Input/ Output)	These are general purpose I/O pins. Their input/output direction is determined by the contents of Data Direction Register (DDR). Port A is selected for write operations when the Chip Enables are active and $\overline{IOW}$ is low and a 0 was previously latched from AD <sub>0</sub> .
A <sub>8-10</sub> (Input)	These are the high order bits of the ROM address. They do not affect I/O operations.		Read operation is selected by either $\overline{IOR}$ low and active Chip Enables and AD <sub>0</sub> low, or IO/ $\overline{M}$ high, $\overline{RD}$ low, active chip enables, and AD <sub>0</sub> low.
$\overline{CE}_1$ CE <sub>2</sub> (Input)	Chip Enable Inputs: $\overline{CE}_1$ is active low and CE <sub>2</sub> is active high. The 8355 can be accessed only when BOTH Chip Enables are active at the time the ALE signal latches them up. If either Chip Enable input is not active, the AD <sub>0-7</sub> and READY outputs will be in a high impedance state.	PB <sub>0-7</sub> (Input/ Output)	This general purpose I/O port is identical to Port A except that it is selected by a 1 latched from AD <sub>0</sub> .
IO/ $\overline{M}$ (Input)	If the latched IO/ $\overline{M}$ is high when $\overline{RD}$ is low, the output data comes from an I/O port. If it is low the output data comes from the ROM.	RESET (Input)	An input high on RESET causes all pins in Port A and B to assume input mode.
$\overline{RD}$ (Input)	If the latched Chip Enables are active when $\overline{RD}$ goes low, the AD <sub>0-7</sub> output buffers are enabled and output either the selected ROM location or I/O port. When both $\overline{RD}$ and $\overline{IOR}$ are high, the AD <sub>0-7</sub> output buffers are 3-state.	$\overline{IOR}$ (Input)	When the Chip Enables are active, a low on $\overline{IOR}$ will output the selected I/O port onto the AD bus. $\overline{IOR}$ low performs the same function as the combination IO/ $\overline{M}$ high and $\overline{RD}$ low. When $\overline{IOR}$ is not used in a system, $\overline{IOR}$ should be tied to V <sub>CC</sub> ("1").
$\overline{IOW}$ (Input)	If the latched Chip Enables are active, a low on $\overline{IOW}$ causes the output port pointed to by the latched value of AD <sub>0</sub> to be written with the data on AD <sub>0-7</sub> . The state of IO/ $\overline{M}$ is ignored.	V <sub>CC</sub> V <sub>SS</sub>	+5 volt supply. Ground Reference.

## FUNCTIONAL DESCRIPTION

### ROM Section

The 8355 contains an 8-bit address latch which allows it to interface directly to MCS-48 and MCS-85 Microcomputers without additional hardware.

The ROM section of the chip is addressed by an 11-bit address and the Chip Enables. The address and levels on the Chip Enable pins are latched into the address latches on the falling edge of ALE. If the latched Chip Enables are active and  $\text{IO}/\overline{\text{M}}$  is low when  $\overline{\text{RD}}$  goes low, the contents of the ROM location addressed by the latched address are put out through  $\text{AD}_{0-7}$  output buffers.

### I/O Section

The I/O section of the chip is addressed by the latched value of  $\text{AD}_{0-1}$ . Two 8-bit Data Direction Registers (DDR) in 8355 determine the input/output status of each pin in the corresponding ports. A "0" in a particular bit position of a DDR signifies that the corresponding I/O port bit is in the input mode. A "1" in a particular bit position signifies that the corresponding I/O port bit is in the output mode. In this manner the I/O ports of the 8355 are bit-by-bit programmable as inputs or outputs. The table summarizes port and DDR designation. DDR's cannot be read.

$\text{AD}_1$	$\text{AD}_0$	Selection
0	0	Port A
0	1	Port B
1	0	Port A Data Direction Register (DDR A)
1	1	Port B Data Direction Register (DDR B)

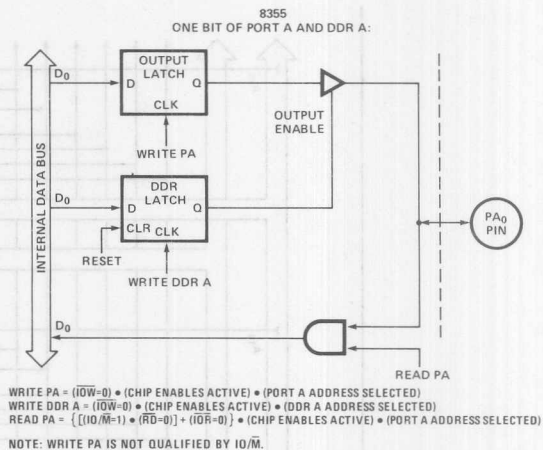
When  $\overline{\text{IOW}}$  goes low and the Chip Enables are active, the data on the  $\text{AD}_{0-7}$  is written into I/O port selected by the latched value of  $\text{AD}_{0-1}$ . During this operation all I/O bits of the selected port are affected, regardless of their I/O mode and the state of  $\text{IO}/\overline{\text{M}}$ . The actual output level does not change until  $\overline{\text{IOW}}$  returns high (glitch free output).

A port can be read out when the latched Chip Enables are active and either  $\overline{\text{RD}}$  goes low with  $\text{IO}/\overline{\text{M}}$  high, or  $\overline{\text{IOR}}$  goes low. Both input and output mode bits of a selected port will appear on lines  $\text{AD}_{0-7}$ .

To clarify the function of the I/O ports and Data Direction Registers, the following diagram shows the configuration of one bit of PORT A and DDR A. The same logic applies to PORT B and DDR B.

Note that hardware RESET or writing a zero to the DDR latch will cause the output latch's output buffer to be disabled, preventing the data in the output latch from being passed through to the pin. This is equivalent to putting the port in the input mode. Note also that the data can be written to the Output Latch even though the Output Buffer has been disabled. This enables a port to be initialized with a value prior to enabling the output.

The diagram also shows that the contents of PORT A and PORT B can be read even when the ports are configured as outputs.



### System Interface with 8085A

A system using the 8355 can use either one of the two I/O Interface techniques:

- Standard I/O
- Memory Mapped I/O

If a standard I/O technique is used, the system can use the feature of both CE and  $\overline{\text{CE}}$ . By using a combination of unused address lines  $\text{A}_{11-15}$  and the Chip Enable inputs, the 8085A system can use up to 5 each 8355's without requiring a CE decoder. See Figure 1.

If a memory mapped I/O approach is used the 8355 will be selected by the combination of both the Chip Enables and  $\text{IO}/\overline{\text{M}}$  using the  $\text{AD}_{8-15}$  address lines. See Figure 2.

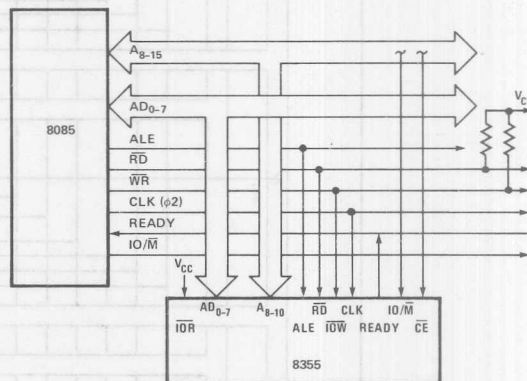
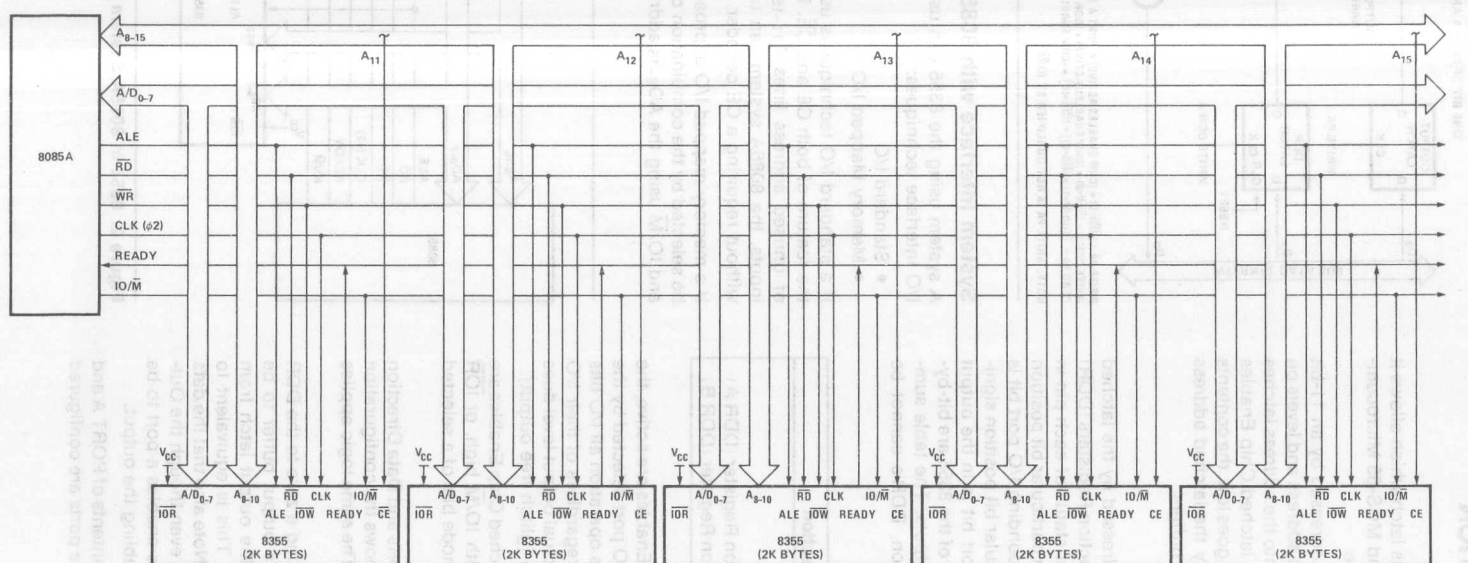


Figure 2. 8355 in 8085A System (Memory-Mapped I/O)



Note: Use  $\overline{CE}$  for the first 8355 in the system, and CE for the other 8355's. Permits up to 5 8355's in a system without CE decoder.

It is not necessary to connect the READY outputs of the 8355's to the READY input of the 8085A unless operating with WAIT states is desired.

Figure 1. 8355 in 8085A System (Standard I/O)

**ABSOLUTE MAXIMUM RATINGS\***

Temperature Under Bias ..... 0°C to +70°C  
 Storage Temperature ..... -65°C to +150°C  
 Voltage on Any Pin  
   With Respect to Ground ..... -0.5V to +7V  
 Power Dissipation ..... 1.5W

\*COMMENT: Stresses above those listed under "Absolute Maximum Ratings" may cause permanent damage to the device. This is a stress rating only and functional operation of the device at these or any other conditions above those indicated in the operational sections of this specification is not implied. Exposure to absolute maximum rating conditions for extended periods may affect device reliability.

**D.C. CHARACTERISTICS** ( $T_A = 0^\circ\text{C}$  to  $70^\circ\text{C}$ ;  $V_{CC} = 5V \pm 5\%$ )

SYMBOL	PARAMETER	MIN.	MAX.	UNITS	TEST CONDITIONS
$V_{IL}$	Input Low Voltage	-0.5	0.8	V	$V_{CC} = 5.0V$
$V_{IH}$	Input High Voltage	2.0	$V_{CC} + 0.5$	V	$V_{CC} = 5.0V$
$V_{OL}$	Output Low Voltage		0.45	V	$I_{OL} = 2mA$
$V_{OH}$	Output High Voltage	2.4		V	$I_{OH} = -400\mu A$
$I_{IL}$	Input Leakage		10	$\mu A$	$V_{IN} = V_{CC}$ to 0V
$I_{LO}$	Output Leakage Current		$\pm 10$	$\mu A$	$0.45V \leq V_{OUT} \leq V_{CC}$
$I_{CC}$	$V_{CC}$ Supply Current		180	mA	

**A.C. CHARACTERISTICS** ( $T_A = 0^\circ\text{C}$  to  $70^\circ\text{C}$ ;  $V_{CC} = 5V \pm 5\%$ )

Symbol	Parameter	8355		8355-2 (Preliminary)		Units
		Min.	Max.	Min.	Max.	
$t_{CYC}$	Clock Cycle Time	320		320		ns
$T_1$	CLK Pulse Width	80		80		ns
$T_2$	CLK Pulse Width	120		120		ns
$t_r, t_f$	CLK Rise and Fall Time		30		30	ns
$t_{AL}$	Address to Latch Set Up Time	50		30		ns
$t_{LA}$	Address Hold Time after Latch	80		30		ns
$t_{LC}$	Latch to READ/WRITE Control	100		40		ns
$t_{RD}$	Valid Data Out Delay from READ Control		170		140	ns
$t_{AD}$	Address Stable to Data Out Valid		400		330	ns
$t_{LL}$	Latch Enable Width	100		70		ns
$t_{RDF}$	Data Bus Float after READ	0	100	0	85	ns
$t_{CL}$	READ/WRITE Control to Latch Enable	20		10		ns
$t_{CC}$	READ/WRITE Control Width	250		200		ns
$t_{DW}$	Data In to Write Set Up Time	150		150		ns
$t_{WD}$	Data In Hold Time After WRITE	10		10		ns
$t_{WP}$	WRITE to Port Output		400		400	ns
$t_{PR}$	Port Input Set Up Time	50		50		ns
$t_{RP}$	Port Input Hold Time	50		50		ns
$t_{RYH}$	READY HOLD Time	0	160	0	160	ns
$t_{ARY}$	ADDRESS (CE) to READY		160		160	ns
$t_{RV}$	Recovery Time Between Controls	300		200		ns
$t_{RDE}$	READ Control to Data Bus Enable	10		10		ns

Note:  $C_{LOAD} = 150pF$

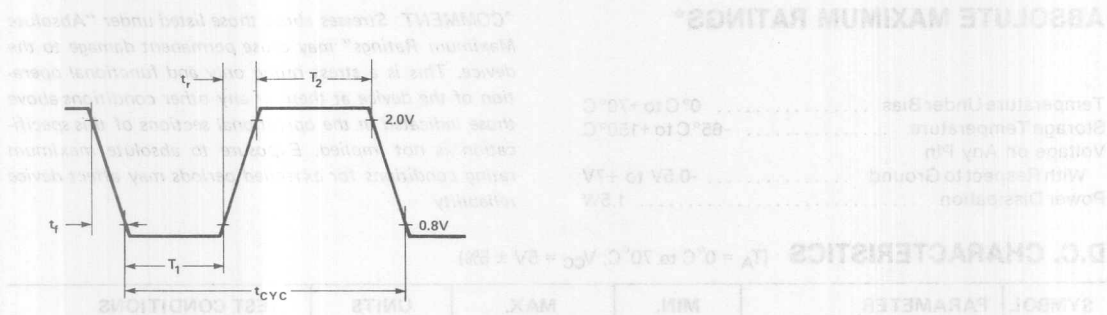
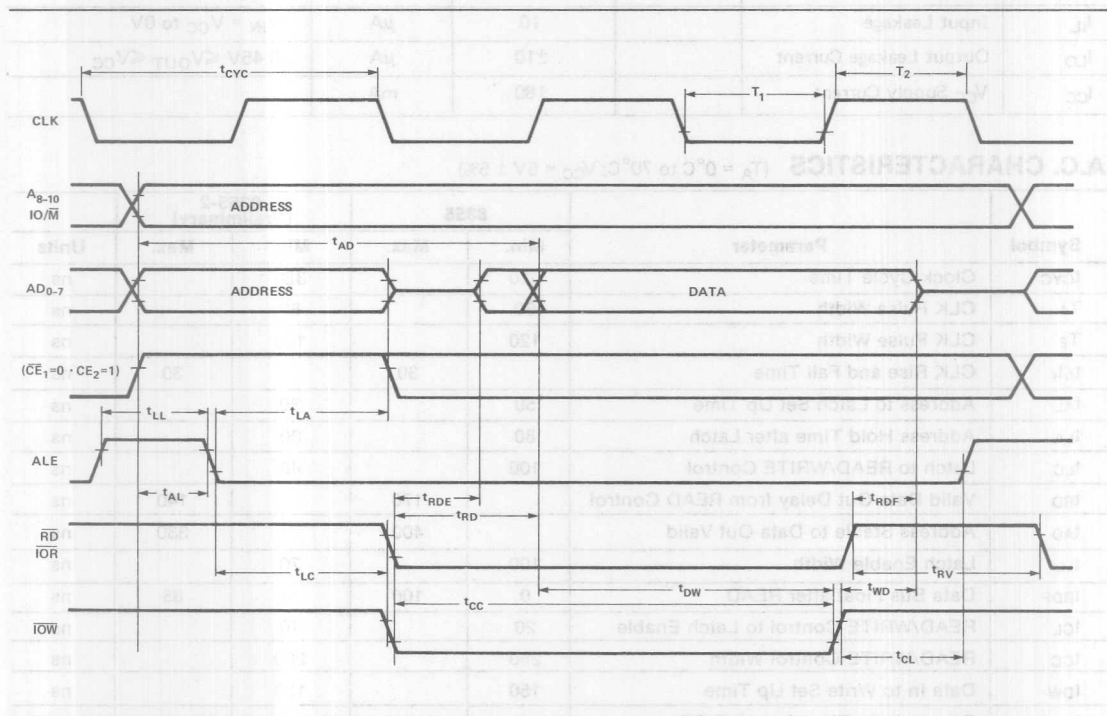
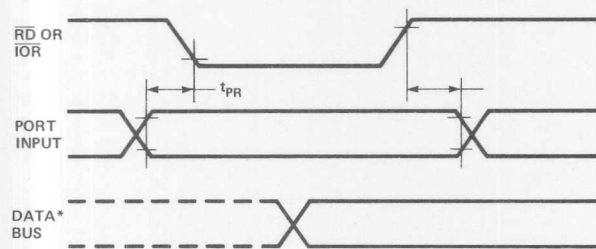


Figure 3. Clock Specification for 8355

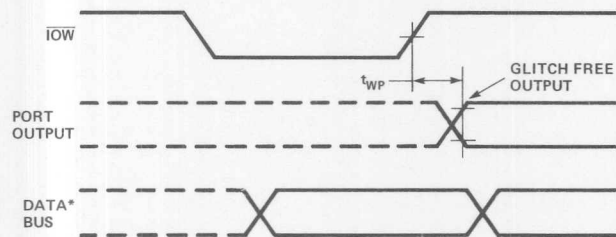


**Figure 4. ROM Read and I/O Read and Write**

## a. Input Mode



## b. Output Mode



\*DATA BUS TIMING IS SHOWN IN FIGURE 4.

Figure 5. I/O Port Timing

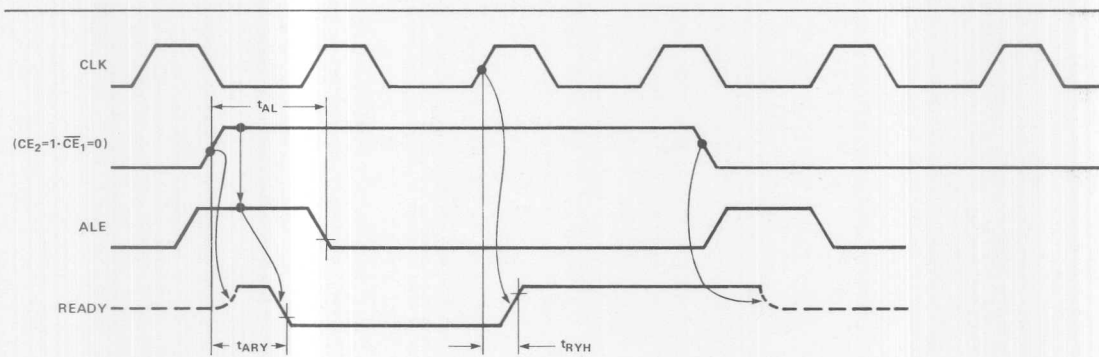


Figure 6. Wait State Timing (Ready = 0)



# 8755A

## 16,384-BIT EPROM WITH I/O

- Directly Compatible with 8085A CPU

■ 2048 Words x 8 Bits

■ Single +5V Power Supply ( $V_{CC}$ )

■ U.V. Erasable and Electrically Reprogrammable

■ Internal Address Latch

■ 2 General Purpose 8-Bit I/O Ports

■ Each I/O Port Line Individually Programmable as Input or Output

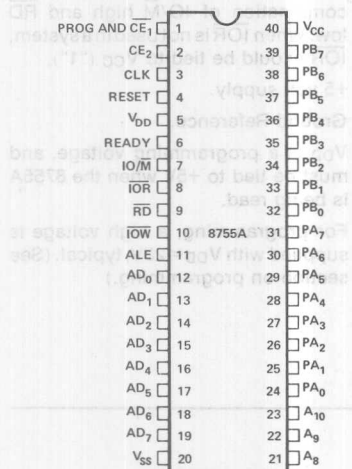
■ Multiplexed Address and Data Bus

■ 40-Pin DIP

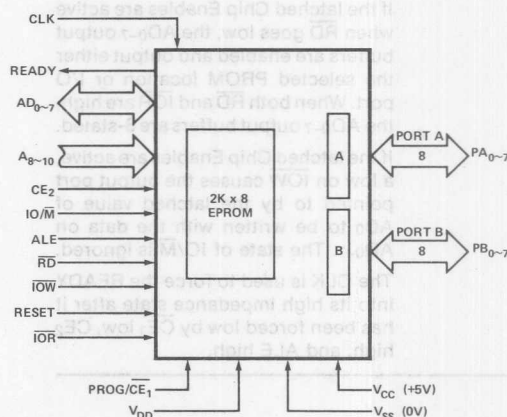
The Intel® 8755A is an erasable and electrically reprogrammable ROM (EPROM) and I/O chip to be used in the MCS-85™ microcomputer system. The EPROM portion is organized as 2048 words by 8 bits. It has a maximum access time of 450 ns to permit use with no wait states in an 8085A CPU.

The I/O portion consists of 2 general purpose I/O ports. Each I/O port has 8 port lines, and each I/O port line is individually programmable as input or output.

### PIN CONFIGURATION



### BLOCK DIAGRAM



ALE (input)	When Address Latch Enable goes high, AD <sub>0-7</sub> , IO/M, A <sub>8-10</sub> , CE <sub>2</sub> , and CE <sub>1</sub> enter the address latches. The signals (AD, IO/M, A <sub>8-10</sub> , CE) are latched in at the trailing edge of ALE.	READY (output)	READY is a 3-state output controlled by CE <sub>2</sub> , $\overline{CE_1}$ , ALE and CLK. READY is forced low when the Chip Enables are active during the time ALE is high, and remains low until the rising edge of the next CLK. (See Figure 6.)
AD <sub>0-7</sub> (input/output)	Bidirectional Address/Data bus. The lower 8-bits of the PROM or I/O address are applied to the bus lines when ALE is high.  During an I/O cycle, Port A or B are selected based on the latched value of AD <sub>0</sub> . If $\overline{RD}$ or $\overline{IOR}$ is low when the latched Chip Enables are active, the output buffers present data on the bus.	PA <sub>0-7</sub> (input/output)	These are general purpose I/O pins. Their input/output direction is determined by the contents of Data Direction Register (DDR). Port A is selected for write operations when the Chip Enables are active and $\overline{IOW}$ is low and a 0 was previously latched from AD <sub>0</sub> , AD <sub>1</sub> .  Read operation is selected by either $\overline{IOR}$ low and active Chip Enables and AD <sub>0</sub> and AD <sub>1</sub> low, or IO/M high, $\overline{RD}$ low, active Chip Enables, and AD <sub>0</sub> and AD <sub>1</sub> low.
A <sub>8-10</sub> (input)	These are the high order bits of the PROM address. They do not affect I/O operations.		
PROG/ $\overline{CE_1}$ CE <sub>2</sub> (input)	Chip Enable Inputs: $\overline{CE_1}$ is active low and CE <sub>2</sub> is active high. The 8755A can be accessed only when BOTH Chip Enables are active at the time the ALE signal latches them up. If either Chip Enable input is not active, the AD <sub>0-7</sub> and READY outputs will be in a high impedance state. $\overline{CE_1}$ is also used as a programming pin. (See section on programming.)	PB <sub>0-7</sub> (input/output)	This general purpose I/O port is identical to Port A except that it is selected by a 1 latched from AD <sub>0</sub> and a 0 from AD <sub>1</sub> .
		RESET (input)	In normal operation, an input high on RESET causes all pins in Ports A and B to assume input mode (clear DDR register).
IO/ $\overline{M}$ (input)	If the latched IO/ $\overline{M}$ is high when $\overline{RD}$ is low, the output data comes from an I/O port. If it is low the output data comes from the PROM.	$\overline{IOR}$ (input)	When the Chip Enables are active, a low on $\overline{IOR}$ will output the selected I/O port onto the AD bus. $\overline{IOR}$ low performs the same function as the combination of IO/M high and $\overline{RD}$ low. When $\overline{IOR}$ is not used in a system, $\overline{IOR}$ should be tied to V <sub>CC</sub> ("1").
$\overline{RD}$ (input)	If the latched Chip Enables are active when $\overline{RD}$ goes low, the AD <sub>0-7</sub> output buffers are enabled and output either the selected PROM location or I/O port. When both $\overline{RD}$ and $\overline{IOR}$ are high, the AD <sub>0-7</sub> output buffers are 3-stated.	V <sub>CC</sub>	+5 volt supply.
		V <sub>SS</sub>	Ground Reference.
$\overline{IOW}$ (input)	If the latched Chip Enables are active, a low on $\overline{IOW}$ causes the output port pointed to by the latched value of AD <sub>0</sub> to be written with the data on AD <sub>0-7</sub> . The state of IO/ $\overline{M}$ is ignored.	V <sub>DD</sub>	V <sub>DD</sub> is a programming voltage, and must be tied to +5V when the 8755A is being read.
CLK (input)	The CLK is used to force the READY into its high impedance state after it has been forced low by CE <sub>1</sub> low, CE <sub>2</sub> high, and ALE high.		For programming, a high voltage is supplied with V <sub>DD</sub> = 25V, typical. (See section on programming.)

## FUNCTIONAL DESCRIPTION

### PROM Section

The 8755A contains an 8-bit address latch which allows it to interface directly to MCS-48 and MCS-85 Microcomputers without additional hardware.

The PROM section of the chip is addressed by the 11-bit address and CE. The address, CE<sub>1</sub> and CE<sub>2</sub> are latched into the address latches on the falling edge of ALE. If the latched Chip Enables are active and IO/M is low when RD goes low, the contents of the PROM location addressed by the latched address are put out on the AD<sub>0-7</sub> lines.

### I/O Section

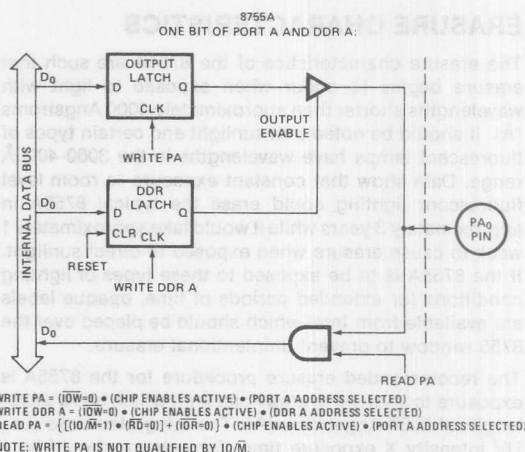
The I/O section of the chip is addressed by the latched value of AD<sub>0-1</sub>. Two 8-bit Data Direction Registers (DDR) in 8755A determine the input/output status of each pin in the corresponding ports. A "0" in a particular bit position of a DDR signifies that the corresponding I/O port bit is in the input mode. A "1" in a particular bit position signifies that the corresponding I/O port bit is in the output mode. In this manner the I/O ports of the 8755A are bit-by-bit programmable as inputs or outputs. The table summarizes port and DDR designation. DDR's cannot be read.

AD <sub>1</sub>	AD <sub>0</sub>	Selection
0	0	Port A
0	1	Port B
1	0	Port A Data Direction Register (DDR A)
1	1	Port B Data Direction Register (DDR B)

When IOW goes low and the Chip Enables are active, the data on the AD is written into I/O port selected by the latched value of AD<sub>0-1</sub>. During this operation all I/O bits of the selected port are affected, regardless of their I/O mode and the state of IO/M. The actual output level does not change until IOW returns high. (glitch free output)

A port can be read out when the latched Chip Enables are active and either RD goes low with IO/M high, or IOR goes low. Both input and output mode bits of a selected port will appear on lines AD<sub>0-7</sub>.

To clarify the function of the I/O Ports and Data Direction Registers, the following diagram shows the configuration of one bit of PORT A and DDR A. The same logic applies to PORT B and DDR B.



Note that hardware RESET or writing a zero to the DDR latch will cause the output latch's output buffer to be disabled, preventing the data in the Output Latch from being passed through to the pin. This is equivalent to putting the port in the input mode. Note also that the data can be written to the Output Latch even though the Output Buffer has been disabled. This enables a port to be initialized with a value prior to enabling the output.

The diagram also shows that the contents of PORT A and PORT B can be read even when the ports are configured as outputs.

TABLE 1. 8755A PROGRAMMING MODULE CROSS REFERENCE

MODULE NAME	USE WITH
UPP 955	UPP(4)
UPP UP2(2)	UPP 855
PROMPT 975	PROMPT 80/85(3)
PROMPT 475	PROMPT 48(1)
NOTES:	
1. Described on p. 13-34 of 1978 Data Catalog.	
2. Special adapter socket.	
3. Described on p. 13-39 of 1978 Data Catalog.	
4. Described on p. 13-71 of 1978 Data Catalog.	

## ERASURE CHARACTERISTICS

The erasure characteristics of the 8755A are such that erasure begins to occur when exposed to light with wavelengths shorter than approximately 4000 Angstroms ( $\text{\AA}$ ). It should be noted that sunlight and certain types of fluorescent lamps have wavelengths in the 3000-4000 $\text{\AA}$  range. Data show that constant exposure to room level fluorescent lighting could erase the typical 8755A in approximately 3 years while it would take approximately 1 week to cause erasure when exposed to direct sunlight. If the 8755A is to be exposed to these types of lighting conditions for extended periods of time, opaque labels are available from Intel which should be placed over the 8755 window to prevent unintentional erasure.

The recommended erasure procedure for the 8755A is exposure to shortwave ultraviolet light which has a wavelength of 2537 Angstroms ( $\text{\AA}$ ). The integrated dose (i.e., UV intensity X exposure time) for erasure should be a minimum of 15W-sec/cm<sup>2</sup>. The erasure time with this dosage is approximately 15 to 20 minutes using an ultraviolet lamp with a 12000 $\mu$ W/cm<sup>2</sup> power rating. The 8755A should be placed within one inch from the lamp tubes during erasure. Some lamps have a filter on their tubes and this filter should be removed before erasure.

## PROGRAMMING

Initially, and after each erasure, all bits of the EPROM portions of the 8755A are in the "1" state. Information is introduced by selectively programming "0" into the desired bit locations. A programmed "0" can only be changed to a "1" by UV erasure.

The 8755A can be programmed on the Intel® Universal PROM Programmer (UPP), and the PROMPT™ 80/85 and PROMPT-48™ design aids. The appropriate programming modules and adapters for use in programming both 8755A's and 8755's are shown in Table 1.

The program mode itself consists of programming a single address at a time, giving a single 50 msec pulse for every address. Generally, it is desirable to have a verify cycle after a program cycle for the same address as shown in the attached timing diagram. In the verify cycle (i.e., normal memory read cycle) 'V<sub>DD</sub>' should be at +5V.

Preliminary timing diagrams and parameter values pertaining to the 8755A programming operation are contained in Figure 7.

## SYSTEM APPLICATIONS

### System Interface with 8085A

A system using the 8755A can use either one of the two I/O Interface techniques:

- Standard I/O
- Memory Mapped I/O

If a standard I/O technique is used, the system can use the feature of both  $\overline{\text{CE}}_2$  and  $\overline{\text{CE}}_1$ . By using a combination of unused address lines A<sub>11-15</sub> and the Chip Enable inputs, the 8085A system can use up to 5 each 8755A's without requiring a CE decoder. See Figure 2.

If a memory mapped I/O approach is used the 8755A will be selected by the combination of both the Chip Enables and I/O/M using the AD<sub>8-15</sub> address lines. See Figure 1.

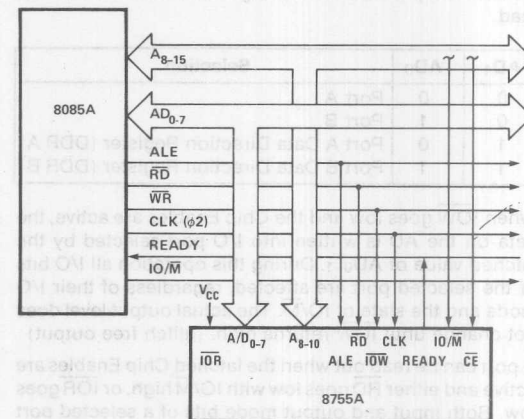
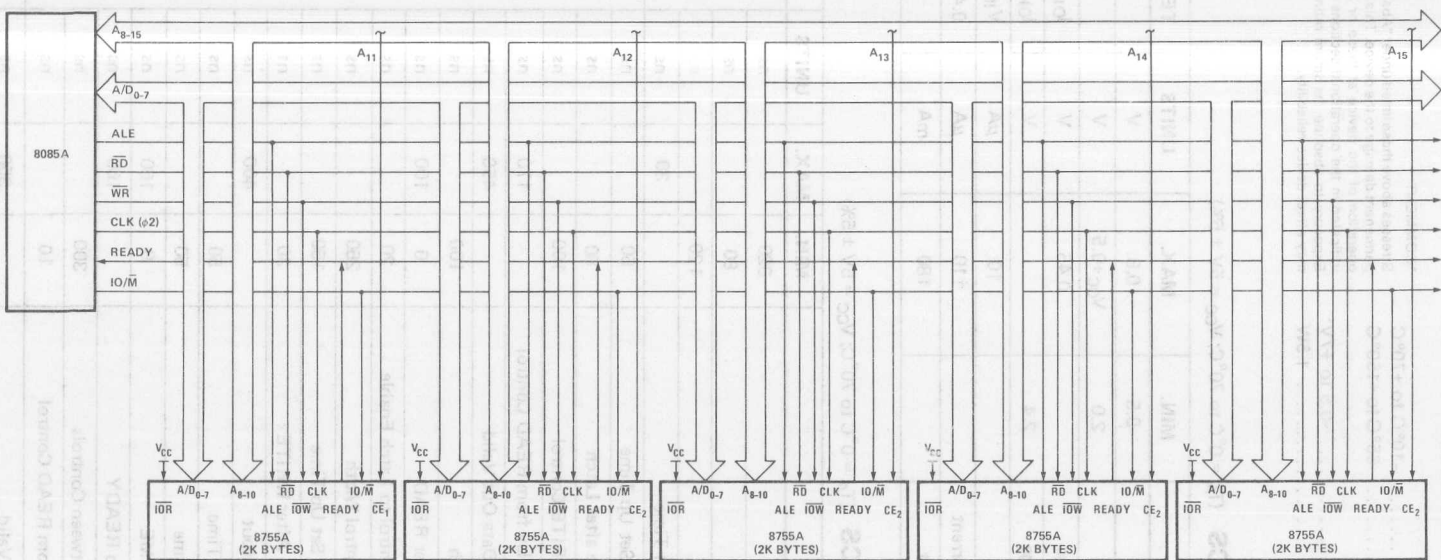


Figure 1. 8755A in 8085A System (Memory-Mapped I/O)



Note: Use  $\overline{CE}_1$  for the first 8755A in the system, and  $\overline{CE}_2$  for the other 8755A's. Permits up to 5-8755A's in a system without CE decoder.

Figure 2. 8755A in 8085A System (Standard I/O)

Temperature Under Bias ..... -10°C to +70°C  
Storage Temperature ..... -65°C to +150°C  
Voltage on Any Pin  
    With Respect to Ground ..... -0.5 to +7V\*  
Power Dissipation ..... 1.5W

\*Except for programming voltage.

#### \*COMMENT

Stresses above those listed under "Absolute Maximum Ratings" may cause permanent damage to the device. This is a stress rating only and functional operation of the device at these or any other conditions above those indicated in the operational sections of this specification is not implied. Exposure to absolute maximum rating conditions for extended periods may affect device reliability.

### D.C. CHARACTERISTICS (T<sub>A</sub> = 0°C to 70°C; V<sub>CC</sub> = 5V ± 5%)

SYMBOL	PARAMETER	MIN.	MAX.	UNITS	TEST CONDITIONS
V <sub>IL</sub>	Input Low Voltage	-0.5	0.8	V	
V <sub>IH</sub>	Input High Voltage	2.0	V <sub>CC</sub> +0.5	V	
V <sub>OL</sub>	Output Low Voltage		0.45	V	I <sub>OL</sub> = 2mA
V <sub>OH</sub>	Output High Voltage	2.4		V	I <sub>OH</sub> = -400μA
I <sub>IL</sub>	Input Leakage		10	μA	V <sub>IN</sub> = V <sub>CC</sub> to 0V
I <sub>LO</sub>	Output Leakage Current		±10	μA	0.45V ≤ V <sub>OUT</sub> ≤ V <sub>CC</sub>
I <sub>CC</sub>	V <sub>CC</sub> Supply Current		180	mA	

### A.C. CHARACTERISTICS (T<sub>A</sub> = 0°C to 70°C; V<sub>CC</sub> = 5V ± 5%)

SYMBOL	PARAMETER	MIN.	MAX.	UNITS	TEST CONDITIONS
t <sub>CYC</sub>	Clock Cycle Time	320		ns	C <sub>LOAD</sub> = 150 pF (See Figure 3)
T <sub>1</sub>	CLK Pulse Width	80		ns	
T <sub>2</sub>	CLK Pulse Width	120		ns	
t <sub>f</sub> , t <sub>r</sub>	CLK Rise and Fall Time		30	ns	
t <sub>AL</sub>	Address to Latch Set Up Time	50		ns	150 pF Load
t <sub>LA</sub>	Address Hold Time after Latch	80		ns	
t <sub>LC</sub>	Latch to READ/WRITE Control	100		ns	
t <sub>RD</sub>	Valid Data Out Delay from READ Control		170	ns	
t <sub>AD</sub>	Address Stable to Data Out Valid		450	ns	
t <sub>LL</sub>	Latch Enable Width	100		ns	
t <sub>RDF</sub>	Data Bus Float after READ	0	100	ns	
t <sub>CL</sub>	READ/WRITE Control to Latch Enable	20		ns	
t <sub>CC</sub>	READ/WRITE Control Width	250		ns	
t <sub>DW</sub>	Data In to WRITE Set Up Time	150		ns	
t <sub>WD</sub>	Data In Hold Time After WRITE	30		ns	
t <sub>WP</sub>	WRITE to Port Output		400	ns	
t <sub>PR</sub>	Port Input Set Up Time	50		ns	
t <sub>RP</sub>	Port Input Hold Time	50		ns	
t <sub>RYH</sub>	READY HOLD TIME	0	160	ns	
t <sub>ARY</sub>	ADDRESS (CE) to READY		160	ns	
t <sub>RV</sub>	Recovery Time between Controls	300		ns	Preliminary
t <sub>RDE</sub>	Data Out Delay from READ Control	10		ns	
t <sub>LD</sub>	ALE to Data Out Valid		350	ns	

## WAVEFORMS

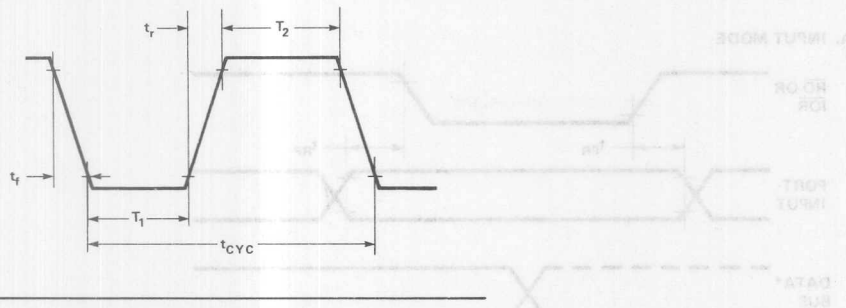


Figure 3. Clock Specification for 8755A

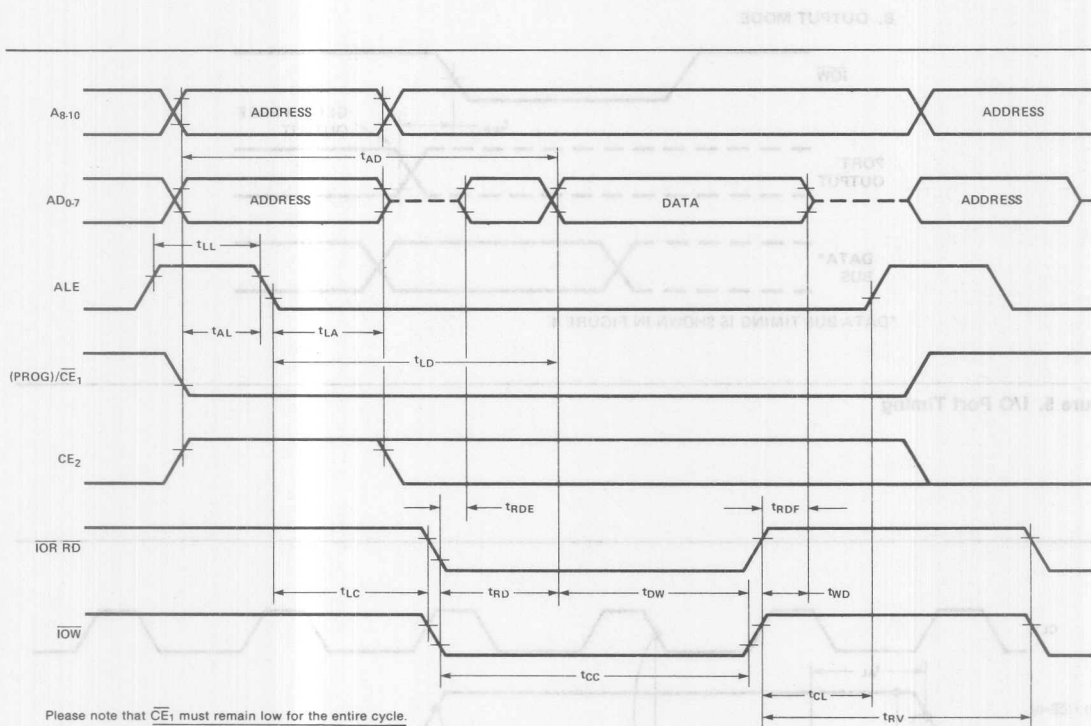


Figure 4. PROM Read, I/O Read and Write Timing

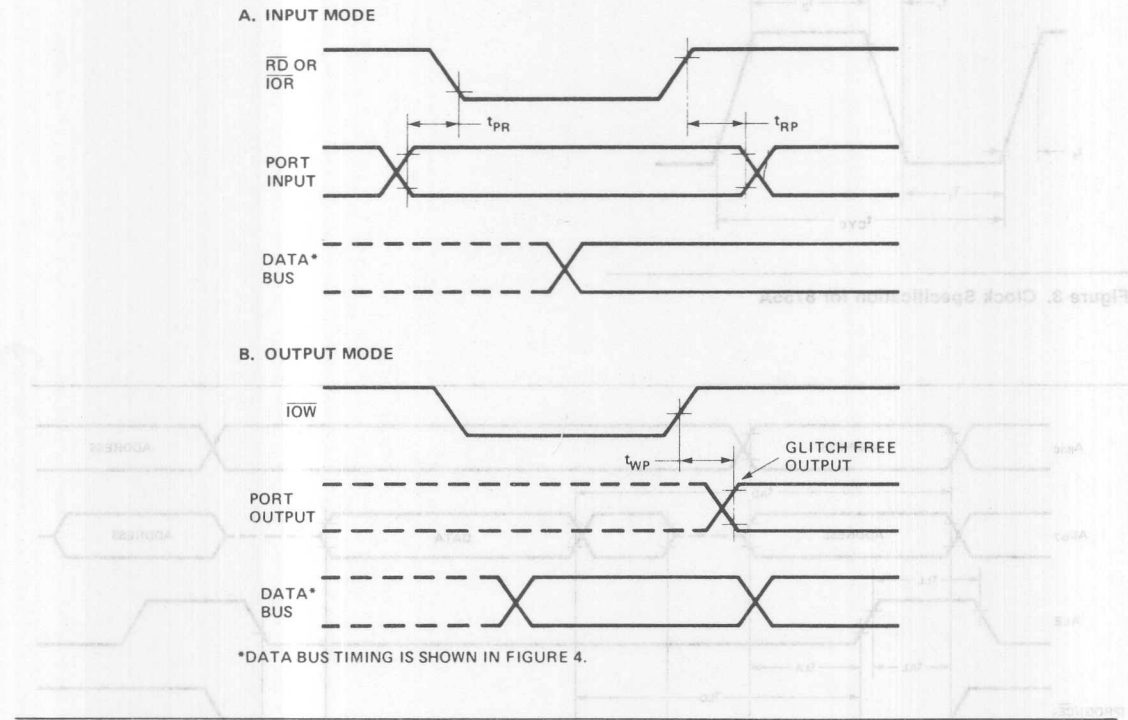


Figure 5. I/O Port Timing

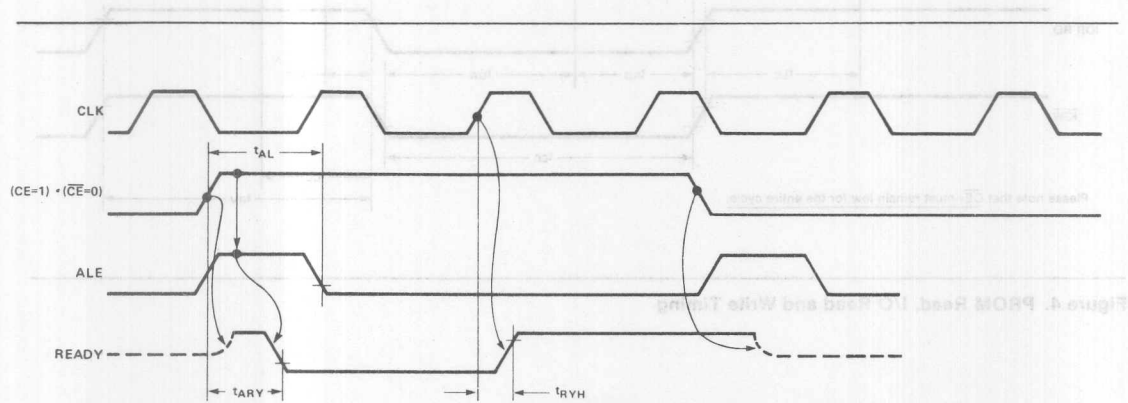


Figure 6. Wait State Timing (READY = 0)

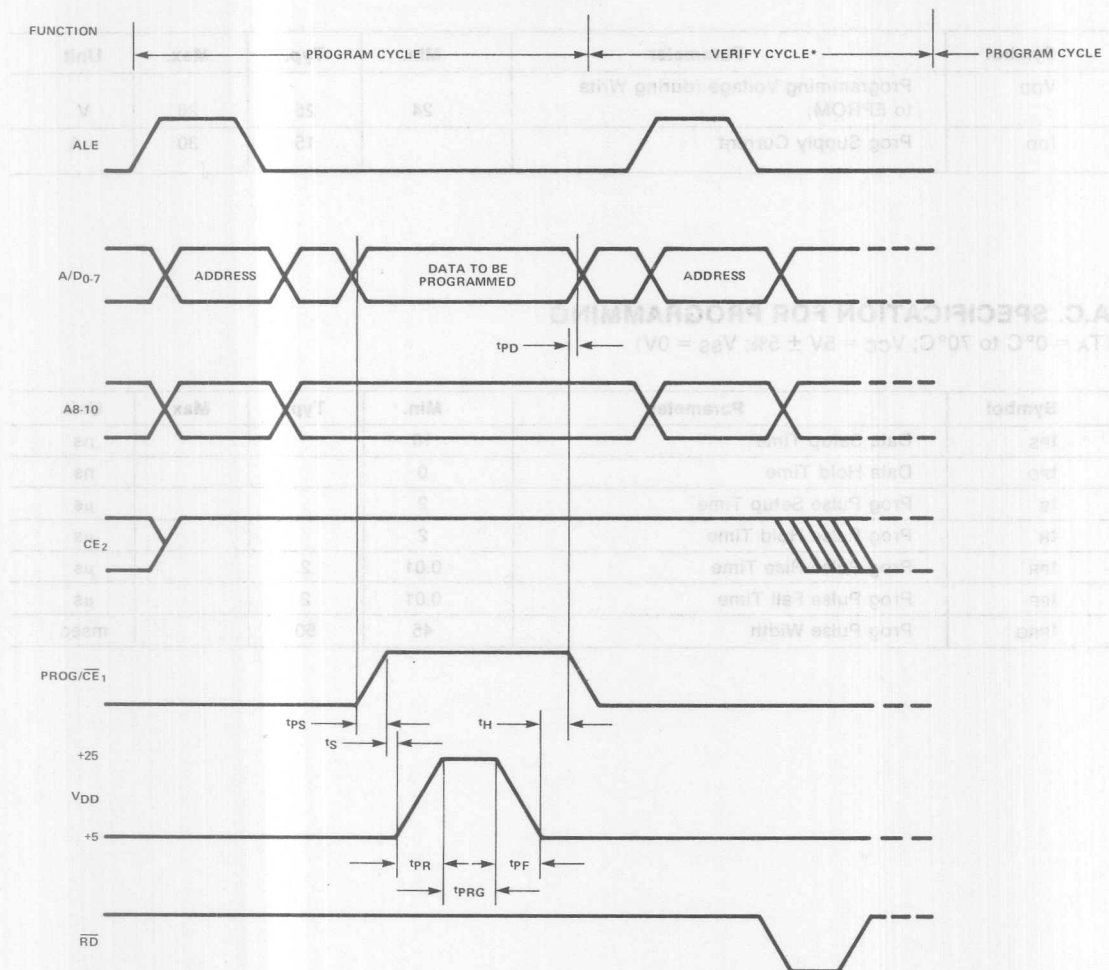
**D.C. SPECIFICATION PROGRAMMING**(T<sub>A</sub> = 0°C to 70°C; V<sub>CC</sub> = 5V ± 5%; V<sub>SS</sub> = 0V)

Symbol	Parameter	Min.	Typ.	Max.	Unit
V <sub>DD</sub>	Programming Voltage (during Write to EPROM)	24	25	26	V
I <sub>DD</sub>	Prog Supply Current		15	30	mA

**A.C. SPECIFICATION FOR PROGRAMMING**(T<sub>A</sub> = 0°C to 70°C; V<sub>CC</sub> = 5V ± 5%; V<sub>SS</sub> = 0V)

Symbol	Parameter	Min.	Typ.	Max.	Unit
t <sub>PS</sub>	Data Setup Time	10			ns
t <sub>PD</sub>	Data Hold Time	0			ns
t <sub>S</sub>	Prog Pulse Setup Time	2			μs
t <sub>H</sub>	Prog Pulse Hold Time	2			μs
t <sub>PR</sub>	Prog Pulse Rise Time	0.01	2		μs
t <sub>PF</sub>	Prog Pulse Fall Time	0.01	2		μs
t <sub>PRG</sub>	Prog Pulse Width	45	50		msec

## WAVEFORMS



\*VERIFY CYCLE IS A REGULAR MEMORY READ CYCLE (WITH V<sub>DD</sub> = +5V FOR 8755A)

Figure 7. 8755A Program Mode Timing Diagram

## CHAPTER 5

**MCS-85**

**System Support  
Components**

**Peripherals**

**Static RAMs**

**ROMs/EPROMs**

**MCS  
85**

**MCS  
85**

**MCS  
85**



# 8205 HIGH SPEED 1 OUT OF 8 BINARY DECODER

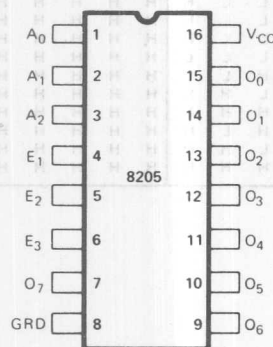
- I/O Port or Memory Selector
- Simple Expansion — Enable Inputs
- High Speed Schottky Bipolar Technology — 18 ns Max Delay
- Directly Compatible with TTL Logic Circuits

- Low Input Load Current — 0.25 mA Max, 1/6 Standard TTL Input Load
- Minimum Line Reflection — Low Voltage Diode Input Clamp
- Outputs Sink 10 mA Min
- 16-Pin Dual In-Line Ceramic or Plastic Package

The Intel® 8205 decoder can be used for expansion of systems which utilize input ports, output ports, and memory components with active low chip select input. When the 8205 is enabled, one of its 8 outputs goes "low", thus a single row of a memory system is selected. The 3-chip enable inputs on the 8205 allow easy system expansion. For very large systems, 8205 decoders can be cascaded such that each decoder can drive 8 other decoders for arbitrary memory expansions.

The 8205 is packaged in a standard 16-pin dual in-line package, and its performance is specified over the temperature range of 0°C to +75°C, ambient. The use of Schottky barrier diode clamped transistors to obtain fast switching speeds results in higher performance than equivalent devices made with a gold diffusion process.

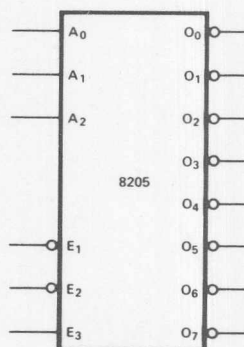
PIN CONFIGURATION



PIN NAMES

A <sub>0</sub> -A <sub>2</sub>	ADDRESS INPUTS
E <sub>1</sub> -E <sub>3</sub>	ENABLE INPUTS
O <sub>0</sub> -O <sub>7</sub>	DECODED OUTPUTS

LOGIC SYMBOL



ADDRESS			ENABLE			OUTPUTS							
A <sub>0</sub>	A <sub>1</sub>	A <sub>2</sub>	E <sub>1</sub>	E <sub>2</sub>	E <sub>3</sub>	0	1	2	3	4	5	6	7
L	L	L	L	L	H	L	H	H	H	H	H	H	H
H	L	L	L	L	H	H	L	H	H	H	H	H	H
L	H	L	L	L	H	H	H	L	H	H	H	H	H
H	H	L	L	L	H	H	H	H	L	H	H	H	H
L	L	H	L	L	H	H	H	H	H	L	H	H	H
H	L	H	L	L	H	H	H	H	H	H	L	H	H
L	H	H	L	L	H	H	H	H	H	H	H	L	H
H	H	H	L	L	H	H	H	H	H	H	H	H	L
X	X	X	L	L	L	H	H	H	H	H	H	H	H
X	X	X	H	L	L	H	H	H	H	H	H	H	H
X	X	X	L	H	L	H	H	H	H	H	H	H	H
X	X	X	H	H	L	H	H	H	H	H	H	H	H
X	X	X	L	H	H	H	H	H	H	H	H	H	H
X	X	X	H	H	H	H	H	H	H	H	H	H	H

## FUNCTIONAL DESCRIPTION

### Decoder

The 8205 contains a one out of eight binary decoder. It accepts a three bit binary code and by gating this input, creates an exclusive output that represents the value of the input code.

For example, if a binary code of 101 was present on the A0, A1 and A2 address input lines, and the device was enabled, an active low signal would appear on the  $\overline{O_5}$  output line. Note that all of the other output pins are sitting at a logic high, thus the decoded output is said to be exclusive. The decoders outputs will follow the truth table shown below in the same manner for all other input variations.

### Enable Gate

When using a decoder it is often necessary to gate the outputs with timing or enabling signals so that the exclusive output of the decoded value is synchronous with the overall system.

The 8205 has a built-in function for such gating. The three enable inputs ( $\overline{E_1}$ ,  $\overline{E_2}$ ,  $\overline{E_3}$ ) are ANDed together and create a single enable signal for the decoder. The combination of both active "high" and active "low" device enable inputs provides the designer with a powerfully flexible gating function to help reduce package count in his system.

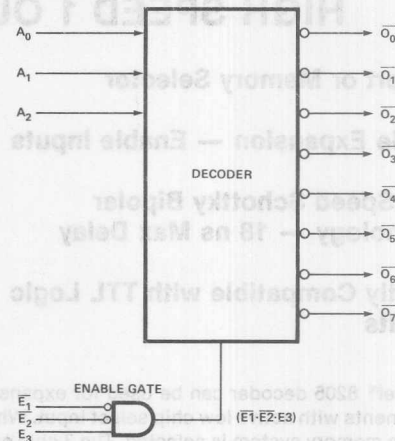


Figure 1. Enable Gate

ADDRESS			ENABLE			OUTPUTS							
A <sub>0</sub>	A <sub>1</sub>	A <sub>2</sub>	E <sub>1</sub>	E <sub>2</sub>	E <sub>3</sub>	0	1	2	3	4	5	6	7
L	L	L	L	L	H	L	H	H	H	H	H	H	H
H	L	L	L	L	H	H	L	H	H	H	H	H	H
L	H	L	L	L	H	H	H	L	H	H	H	H	H
H	H	L	L	L	H	H	H	H	L	H	H	H	H
L	L	H	L	L	H	H	H	H	H	L	H	H	H
H	L	H	L	L	H	H	H	H	H	H	L	H	H
L	H	H	L	L	H	H	H	H	H	H	H	L	H
H	H	H	L	L	H	H	H	H	H	H	H	H	L
X	X	X	L	L	L	H	H	H	H	H	H	H	H
X	X	X	H	L	L	H	H	H	H	H	H	H	H
X	X	X	L	H	L	H	H	H	H	H	H	H	H
X	X	X	H	H	L	H	H	H	H	H	H	H	H
X	X	X	L	H	H	H	H	H	H	H	H	H	H
X	X	X	H	H	H	H	H	H	H	H	H	H	H

## APPLICATIONS OF THE 8205

The 8205 can be used in a wide variety of applications in microcomputer systems. I/O ports can be decoded from the address bus, chip select signals can be generated to select memory devices and the type of machine state such as in 8008 systems can be derived from a simple decoding of the state lines (S0, S1, S2) of the 8008 CPU.

### I/O Port Decoder

Shown in the figure below is a typical application of the 8205. Address input lines are decoded by a group of 8205s (3). Each input has a binary weight. For example, A0 is assigned a value of 1 and is the LSB; A4 is assigned a value of 16 and is the MSB. By connecting them to the decoders as shown, an active low signal that is exclusive in nature and represents the value of the input address lines, is available at the outputs of the 8205s.

This circuit can be used to generate enable signals for I/O ports or any other decoder related application.

Note that no external gating is required to decode up to 24 exclusive devices and that a simple addition of an inverter or two will allow expansion to even larger decoder networks.

### Chip Select Decoder

Using a very similar circuit to the I/O port decoder, an ar-

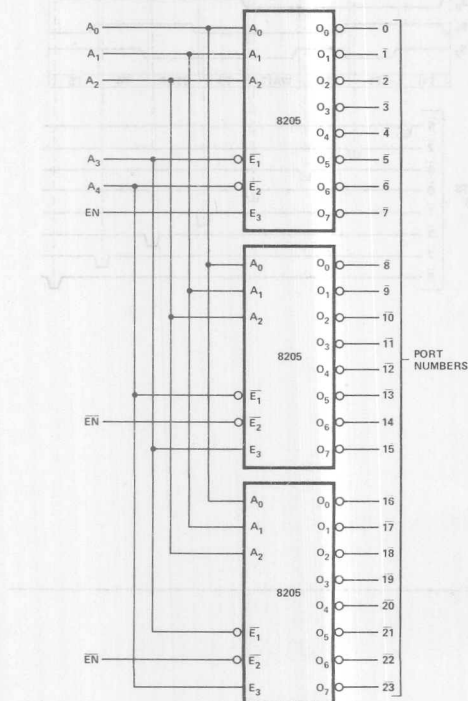


Figure 2. I/O Port Decoder

ray of 8205s can be used to create a simple interface to a 24K memory system.

The memory devices used can be either ROM or RAM and are 1K in storage capacity. 8308s and 8102s are the devices typically used for this application. This type of memory device has ten (10) address inputs and an active "low" chip select (CS). The lower order address bits A0-A9 which come from the microprocessor are "bussed" to all memory elements and the chip select to enable a specific device or group of devices comes from the array of 8205s. The output of the 8205 is active low so it is directly compatible with the memory components.

Basic operation is that the CPU issues an address to identify a specific memory location in which it wishes to "write" or "read" data. The most significant address bits A10-A14 are decoded by the array of 8205s and an exclusive, active low, chip select is generated that enables a specific memory device. The least significant address bits A0-A9 identify a specific location within the selected device. Thus, all addresses throughout the entire memory array are exclusive in nature and are non-redundant.

This technique can be expanded almost indefinitely to support even larger systems with the addition of a few inverters and an extra decoder (8205).

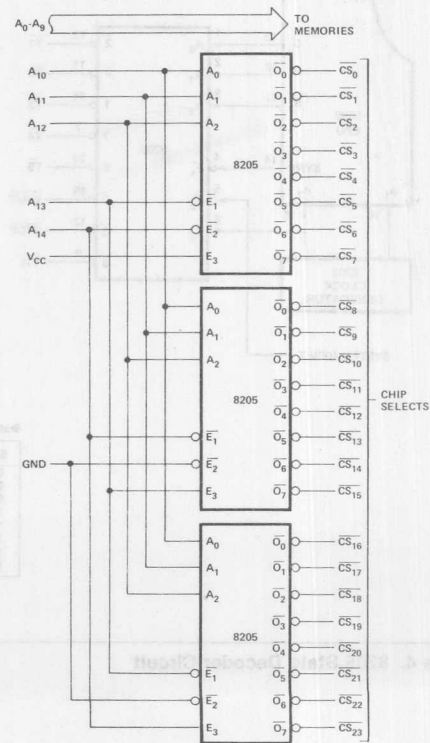


Figure 3. 32K Memory Interface

### Logic Element Example

Probably the most overlooked application of the 8205 is that of a general purpose logic element. Using the "on-chip" enabling gate, the 8205 can be configured to gate its decoded outputs with system timing signals and generate strobes that can be directly connected to latches, flip-flops and one-shots that are used throughout the system.

An excellent example of such an application is the "state decoder" in an 8008 CPU based system. The 8008 CPU issues three bits of information (S<sub>0</sub>, S<sub>1</sub>, S<sub>2</sub>) that indicate the nature of the data on the Data Bus during each machine state. Decoding of these signals is vital to generate strobes that can load the address latches, control bus discipline and general machine functions.

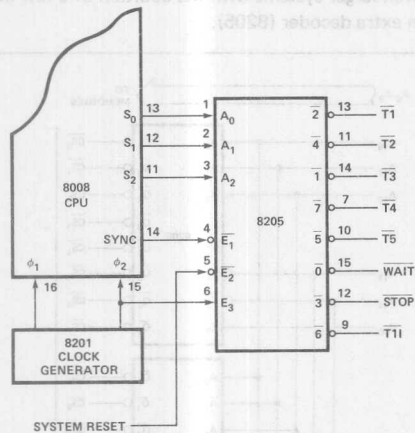
In the figure below a circuit is shown using the 8205 as the "state decoder" for an 8008 CPU that not only decodes the S<sub>0</sub>, S<sub>1</sub>, S<sub>2</sub> outputs but gates these signals with the clock (phase 2) and the SYNC output of the 8008 CPU. The T<sub>1</sub>

and T<sub>2</sub> decoded strobes can connect directly to devices like 8212s for latching the address information. The other decoded strobes can be used to generate signals to control the system data bus, memory timing functions and interrupt structure. RESET is connected to the enable gate so that strobes are not generated during system reset, eliminating accidental loading.

The power of such a circuit becomes evident when a single decoded strobe is logically broken down. Consider T<sub>1</sub> output, the boolean equation for it would be:

$$T_1 = (\overline{S_0} \cdot S_1 \cdot \overline{S_2}) \cdot (\overline{SYNC} \cdot \text{Phase 2} \cdot \overline{\text{Reset}})$$

A six input NAND gate plus a few inverters would be needed to implement this function. The seven remaining outputs would need a similar circuit to duplicate their function, obviously a substantial savings in components can be achieved when using such a technique.



State Control Coding

S <sub>0</sub>	S <sub>1</sub>	S <sub>2</sub>	STATE
0	1	0	T1
0	1	1	T1I
0	0	1	T2
0	0	0	WAIT
1	0	0	T3
1	1	0	STOP
1	1	1	T4
1	0	1	T5

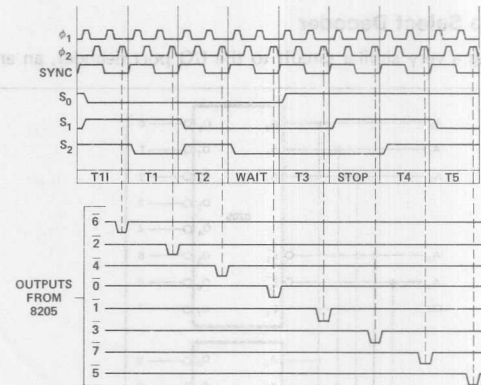


Figure 4. 8205 State Decoder Circuit

## ABSOLUTE MAXIMUM RATINGS\*

Temperature Under Bias:	Ceramic	-65°C to +125°C
	Plastic	-65°C to +75°C
Storage Temperature		-65°C to +160°C
All Output or Supply Voltages		-0.5 to +7 Volts
All Input Voltages		-1.0 to +5.5 Volts
Output Currents		125 mA

## \*COMMENT

Stresses above those listed under "Absolute Maximum Rating" may cause permanent damage to the device. This is a stress rating only and functional operation of the device at these or at any other condition above those indicated in the operational sections of this specification is not implied. Exposure to absolute maximum rating conditions for extended periods may affect device reliability.

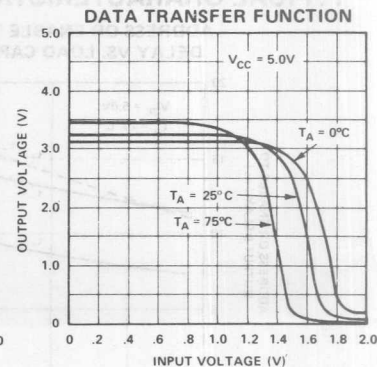
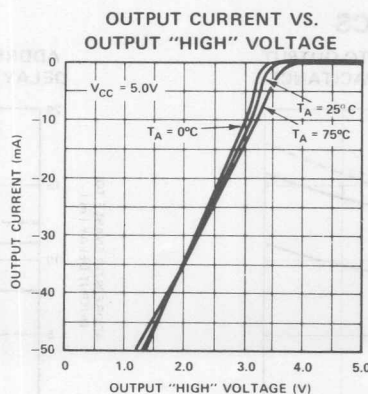
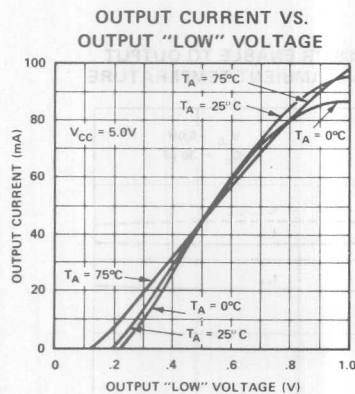
## D.C. CHARACTERISTICS

$T_A = 0^\circ\text{C}$  to  $+75^\circ\text{C}$ ,  $V_{CC} = 5\text{V} \pm 5\%$

8205

SYMBOL	PARAMETER	LIMIT		UNIT	TEST CONDITIONS
		MIN.	MAX.		
$I_F$	INPUT LOAD CURRENT		-0.25	mA	$V_{CC} = 5.25\text{V}$ , $V_F = 0.45\text{V}$
$I_R$	INPUT LEAKAGE CURRENT		10	$\mu\text{A}$	$V_{CC} = 5.25\text{V}$ , $V_R = 5.25\text{V}$
$V_C$	INPUT FORWARD CLAMP VOLTAGE		-1.0	V	$V_{CC} = 4.75\text{V}$ , $I_C = -5.0\text{mA}$
$V_{OL}$	OUTPUT "LOW" VOLTAGE		0.45	V	$V_{CC} = 4.75\text{V}$ , $I_{OL} = 10.0\text{mA}$
$V_{OH}$	OUTPUT HIGH VOLTAGE	2.4		V	$V_{CC} = 4.75\text{V}$ , $I_{OH} = -1.5\text{mA}$
$V_{IL}$	INPUT "LOW" VOLTAGE		0.85	V	$V_{CC} = 5.0\text{V}$
$V_{IH}$	INPUT "HIGH" VOLTAGE	2.0		V	$V_{CC} = 5.0\text{V}$
$I_{SC}$	OUTPUT HIGH SHORT CIRCUIT CURRENT	-40	-120	mA	$V_{CC} = 5.0\text{V}$ , $V_{OUT} = 0\text{V}$
$V_{OX}$	OUTPUT "LOW" VOLTAGE @ HIGH CURRENT		0.8	V	$V_{CC} = 5.0\text{V}$ , $I_{OX} = 40\text{mA}$
$I_{CC}$	POWER SUPPLY CURRENT		70	mA	$V_{CC} = 5.25\text{V}$

## TYPICAL CHARACTERISTICS



## SWITCHING CHARACTERISTICS

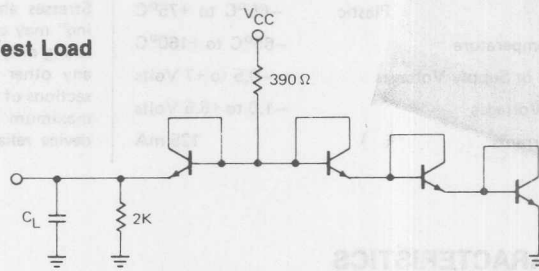
## Conditions of Test:

Input pulse amplitudes: 2.5V

Input rise and fall times: 5 nsec  
between 1V and 2V

Measurements are made at 1.5V

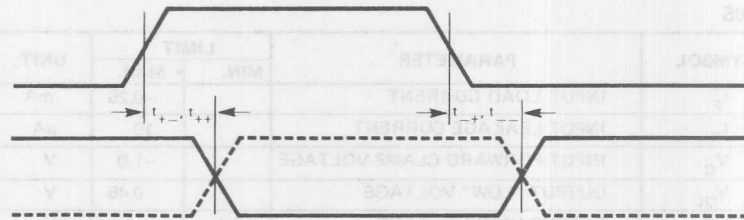
## Test Load

All Transistors 2N2369 or Equivalent.  $C_L = 30$  pF

## Test Waveforms

ADDRESS OR ENABLE  
INPUT PULSE

OUTPUT



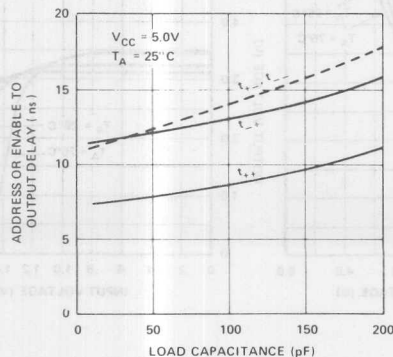
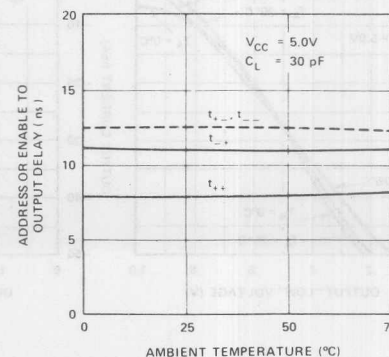
## A.C. CHARACTERISTICS

 $T_A = 0^{\circ}\text{C}$  to  $+75^{\circ}\text{C}$ ,  $V_{CC} = 5\text{V} \pm 5\%$  unless otherwise specified.

SYMBOL	PARAMETER	MAX. LIMIT	UNIT	TEST CONDITIONS
$t_{++}$	ADDRESS OR ENABLE TO OUTPUT DELAY	18	ns	$f = 1\text{ MHz}$ , $V_{CC} = 0\text{V}$ $V_{BIAS} = 2.0\text{V}$ , $T_A = 25^{\circ}\text{C}$
$t_{-+}$		18	ns	
$t_{+-}$		18	ns	
$t_{--}$		18	ns	
$C_{IN}^{(1)}$	INPUT CAPACITANCE	4(typ.) 5(typ.)	pF	
	P8205 C8205			

1. This parameter is periodically sampled and is not 100% tested.

## TYPICAL CHARACTERISTICS

ADDRESS OR ENABLE TO OUTPUT  
DELAY VS. LOAD CAPACITANCEADDRESS OR ENABLE TO OUTPUT  
DELAY VS. AMBIENT TEMPERATURE

# 8212

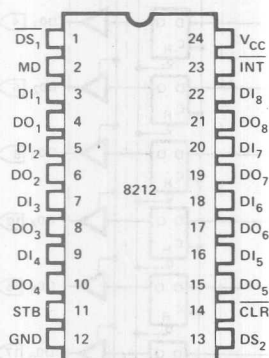
## 8-BIT INPUT/OUTPUT PORT

- Fully Parallel 8-Bit Data Register and Buffer
- Service Request Flip-Flop for Interrupt Generation
- Low Input Load Current — .25mA Max.
- Three State Outputs
- Outputs Sink 15mA
- 3.65V Output High Voltage for Direct Interface to 8008, 8080A, or 8085A CPU
- Asynchronous Register Clear
- Replaces Buffers, Latches and Multiplexers in Microcomputer Systems
- Reduces System Package Count

The 8212 input/output port consists of an 8-bit latch with 3-state output buffers along with control and device selection logic. Also included is a service request flip-flop for the generation and control of interrupts to the microprocessor.

The device is multimode in nature. It can be used to implement latches, gated buffers or multiplexers. Thus, all of the principal peripheral and input/output functions of a microcomputer system can be implemented with this device.

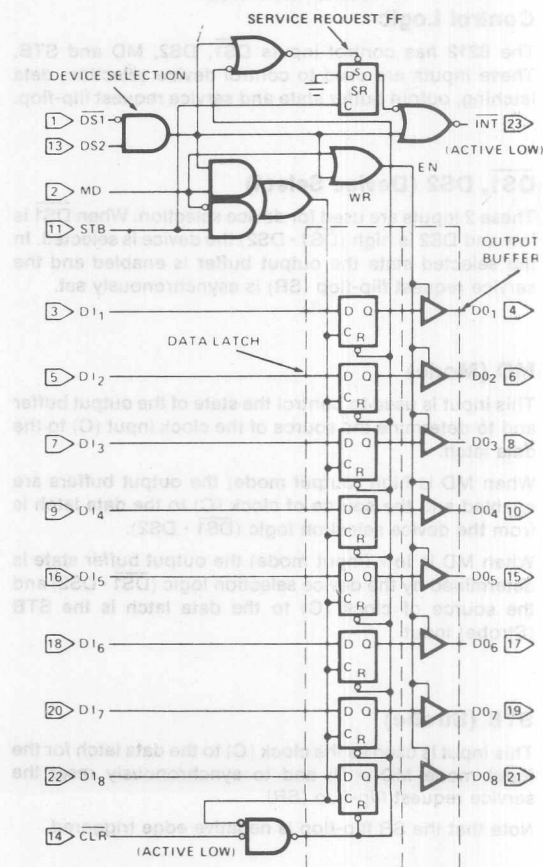
### PIN CONFIGURATION



### PIN NAMES

DI <sub>1</sub> -DI <sub>8</sub>	DATA IN
DO <sub>1</sub> -DO <sub>8</sub>	DATA OUT
DS <sub>1</sub> -DS <sub>2</sub>	DEVICE SELECT
MD	MODE
STB	STROBE
INT	INTERRUPT (ACTIVE LOW)
CLR	CLEAR (ACTIVE LOW)

### LOGIC DIAGRAM



The 8 flip-flops that make up the data latch are of a "D" type design. The output (Q) of the flip-flop will follow the data input (D) while the clock input (C) is high. Latching will occur when the clock (C) returns low.

The latched data is cleared by an asynchronous reset input (CLR). (Note: Clock (C) Overrides Reset (CLR).)

## Output Buffer

The outputs of the data latch (Q) are connected to 3-state, non-inverting output buffers. These buffers have a common control line (EN); this control line either enables the buffer to transmit the data from the outputs of the data latch (Q) or disables the buffer, forcing the output into a high impedance state. (3-state)

The high-impedance state allows the designer to connect the 8212 directly onto the microprocessor bi-directional data bus.

## Control Logic

The 8212 has control inputs  $\overline{DS1}$ , DS2, MD and STB. These inputs are used to control device selection, data latching, output buffer state and service request flip-flop.

## $\overline{DS1}$ , DS2 (Device Select)

These 2 inputs are used for device selection. When  $\overline{DS1}$  is low and DS2 is high ( $\overline{DS1} \cdot DS2$ ) the device is selected. In the selected state the output buffer is enabled and the service request flip-flop (SR) is asynchronously set.

## MD (Mode)

This input is used to control the state of the output buffer and to determine the source of the clock input (C) to the data latch.

When MD is high (output mode) the output buffers are enabled and the source of clock (C) to the data latch is from the device selection logic ( $\overline{DS1} \cdot DS2$ ).

When MD is low (input mode) the output buffer state is determined by the device selection logic ( $\overline{DS1} \cdot DS2$ ) and the source of clock (C) to the data latch is the STB (Strobe) input.

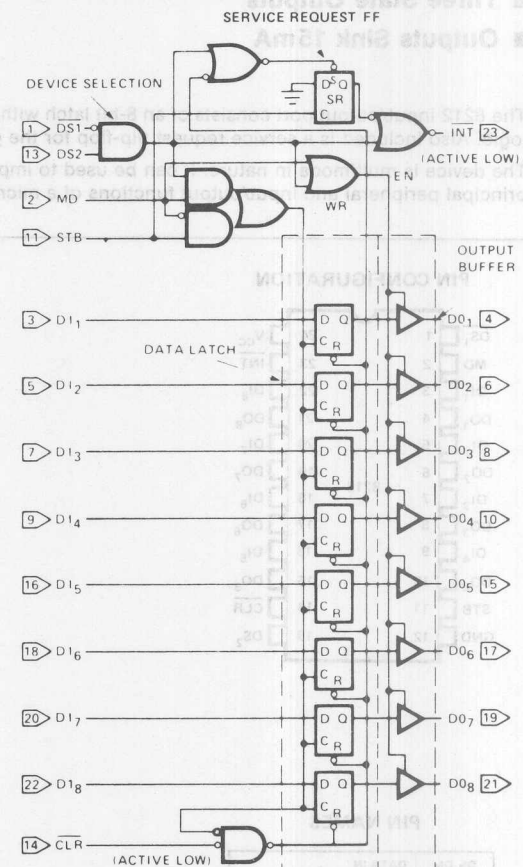
## STB (Strobe)

This input is used as the clock (C) to the data latch for the input mode ( $MD = 0$ ) and to synchronously reset the service request flip-flop (SR).

Note that the SR flip-flop is negative edge triggered.

interrupts in microcomputer systems. It is asynchronously set by the  $\overline{CLR}$  input (active low). When the (SR) flip-flop is set it is in the non-interrupting state.

The output of the (SR) flip-flop (Q) is connected to an inverting input of a "NOR" gate. The other input to the "NOR" gate is non-inverting and is connected to the device selection logic ( $\overline{DS1} \cdot DS2$ ). The output of the "NOR" gate (INT) is active low (interrupting state) for connection to active low input priority generating circuits.



STB	MD	( $\overline{DS1} \cdot DS2$ )	DATA OUT EQUALS	CLR	( $\overline{DS1} \cdot DS2$ )	STB	*SR	INT
0	0	0	3 STATE	0	0	0	1	1
1	0	0	3 STATE	0	1	0	1	0
0	1	0	DATA LATCH	1	1	0	0	0
1	1	0	DATA LATCH	1	1	0	1	0
0	0	1	DATA LATCH	1	0	0	1	0
1	0	1	DATA IN	1	0	0	1	1
0	1	1	DATA IN	1	1	0	1	0
1	1	1	DATA IN	1	1	0	1	0

CLR - RESETS DATA LATCH  
SETS SR FLIP-FLOP  
(NO EFFECT ON OUTPUT BUFFER)

\*INTERNAL SR FLIP-FLOP

## Applications of the 8212 — For Microcomputer Systems

- I Basic Schematic Symbol
- II Gated Buffer
- III Bi-Directional Bus Driver
- IV Interrupting Input Port

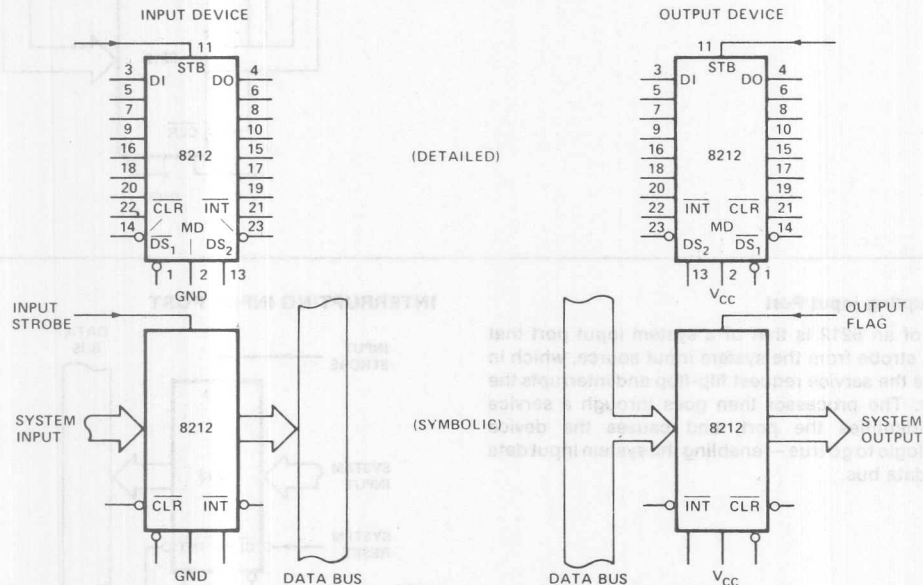
- V Interrupt Instruction Port
- VI Output Port
- VII 8080A Status Latch
- VIII 8085A Address Latch

### 1. Basic Schematic Symbols

Two examples of ways to draw the 8212 on system schematics — (1) the top being the detailed view showing pin numbers, and (2) the bottom being the symbolic view

showing the system input or output as a system bus (bus containing 8 parallel lines). The output to the data bus is symbolic in referencing 8 parallel lines.

### BASIC SCHEMATIC SYMBOLS



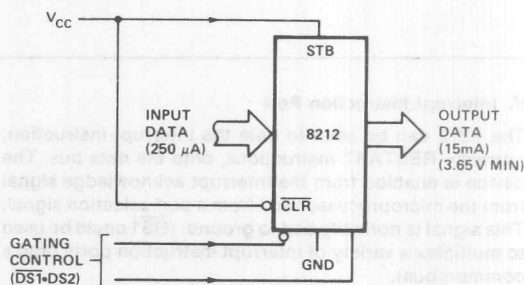
### II. Gated Buffer (3-State)

The simplest use of the 8212 is that of a gated buffer. By tying the mode signal low and the strobe input high, the data latch is acting as a straight through gate. The output buffers are then enabled from the device selection logic DS1 and DS2.

When the device selection logic is false, the outputs are 3-state.

When the device selection logic is true, the input data from the system is directly transferred to the output. The input data load is 250 micro amps. The output data can sink 15 milli amps. The minimum high output is 3.65 volts.

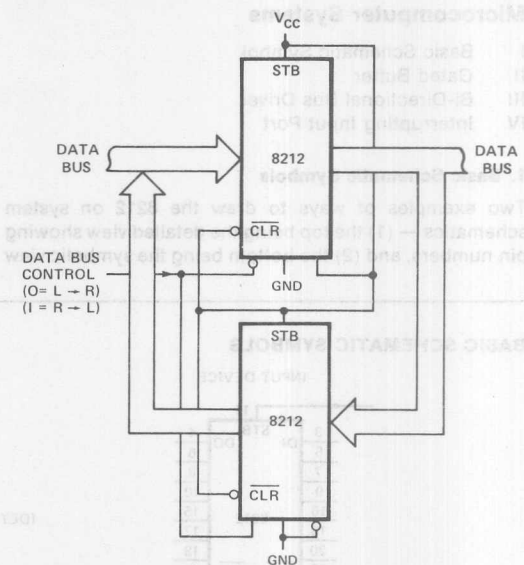
### GATED BUFFER



### III. Bi-Directional Bus Driver

A pair of 8212's wired (back-to-back) can be used as a symmetrical drive, bi-directional bus driver. The devices are controlled by the data bus input control which is connected to  $\overline{DS1}$  on the first 8212 and to  $DS2$  on the second. One device is active, and acting as a straight through buffer the other is in 3-state mode. This is a very useful circuit in small system design.

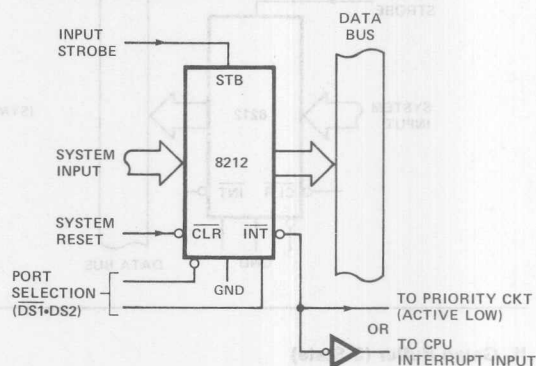
### BI-DIRECTIONAL BUS DRIVER



### IV. Interrupting Input Port

This use of an 8212 is that of a system input port that accepts a strobe from the system input source, which in turn clears the service request flip-flop and interrupts the processor. The processor then goes through a service routine, identifies the port, and causes the device selection logic to go true — enabling the system input data onto the data bus.

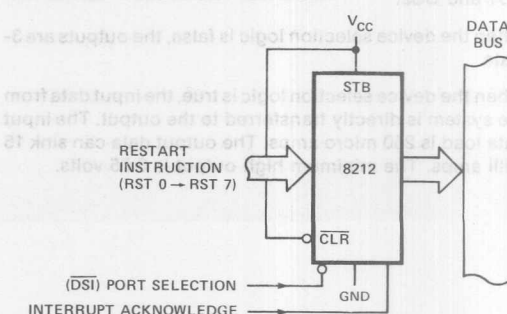
### INTERRUPTING INPUT PORT



### V. Interrupt Instruction Port

The 8212 can be used to gate the interrupt instruction, normally RESTART instructions, onto the data bus. The device is enabled from the interrupt acknowledge signal from the microprocessor and from a port selection signal. This signal is normally tied to ground. ( $\overline{DS1}$  could be used to multiplex a variety of interrupt instruction ports onto a common bus).

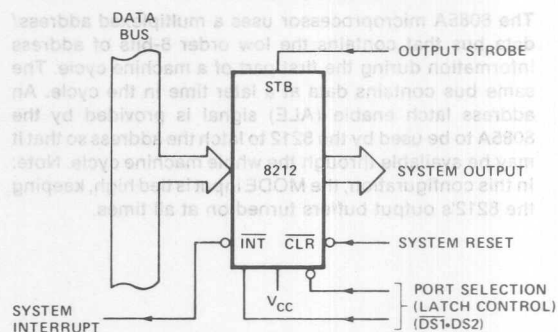
### INTERRUPT INSTRUCTION PORT



## VI. Output Port (With Hand-Shaking)

The 8212 can be used to transmit data from the data bus to a system output. The output strobe could be a hand-shaking signal such as "reception of data" from the device that the system is outputting to. It in turn, can interrupt the system signifying the reception of data. The selection of the port comes from the device selection logic. (DS1 · DS2)

### OUTPUT PORT (WITH HAND-SHAKING)

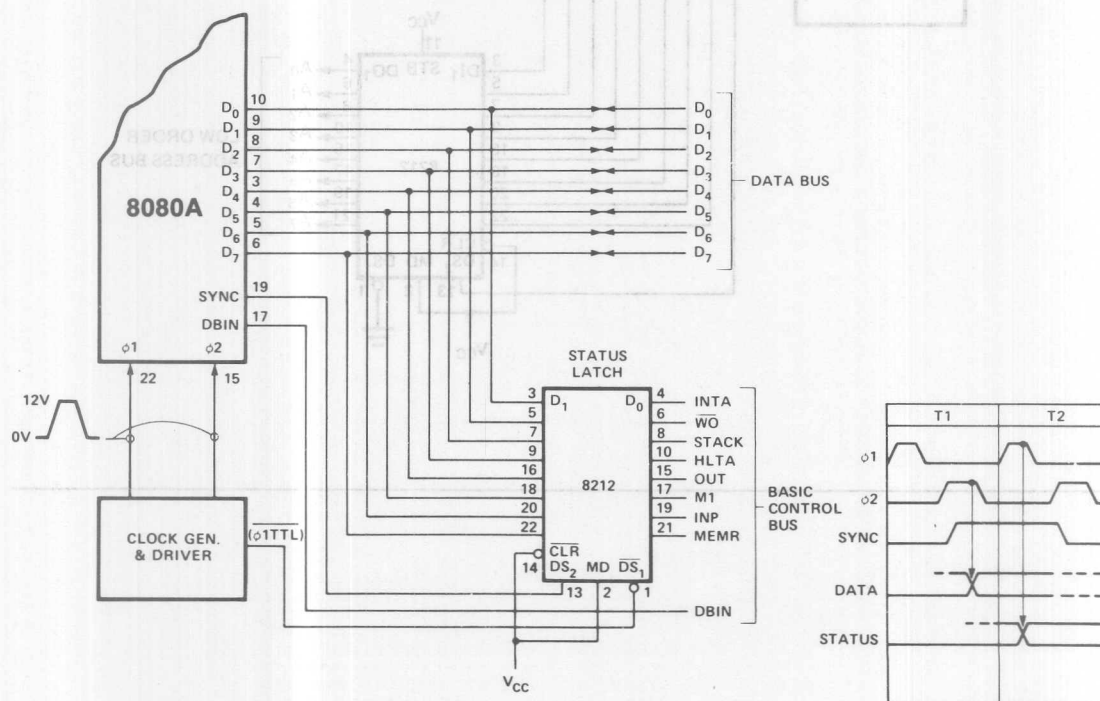


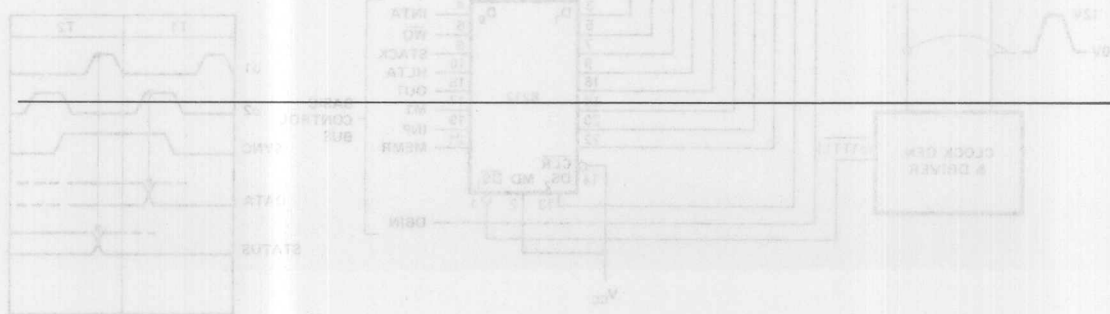
## VII. 8080A Status Latch

Here the 8212 is used as the status latch for an 8080A microcomputer system. The input to the 8212 latch is directly from the 8080A data bus. Timing shows that when the SYNC signal is true, which is connected to the DS2 input and the phase 1 signal is true, which is a TTL level coming from the clock generator; then, the status data will be latched into the 8212.

Note: The mode signal is tied high so that the output on the latch is active and enabled all the time.

It is shown that the two areas of concern are the bi-directional data bus of the microprocessor and the control bus.





## ABSOLUTE MAXIMUM RATINGS\*

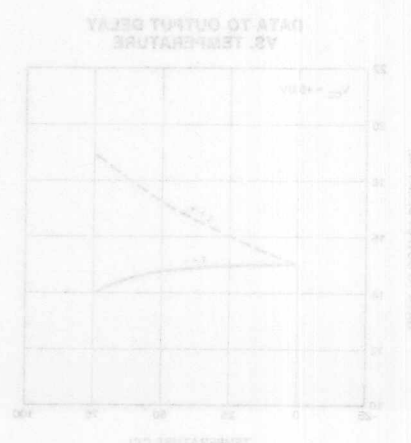
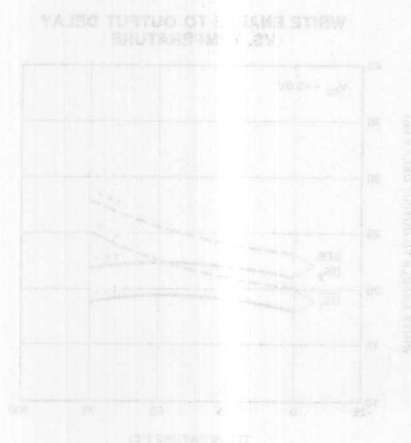
Temperature Under Bias Plastic ..... 0°C to +70°C  
 Storage Temperature ..... -65°C to +160°C  
 All Output or Supply Voltages ..... -0.5 to +7 Volts  
 All Input Voltages ..... -1.0 to 5.5 Volts  
 Output Currents ..... 100mA

## \*COMMENT

Stresses above those listed under "Absolute Maximum Ratings" may cause permanent damage to the device. This is a stress rating only and functional operation of the device at these or any other conditions above those indicated in the operational sections of this specification is not implied. Exposure to absolute maximum rating conditions for extended periods may affect device reliability.

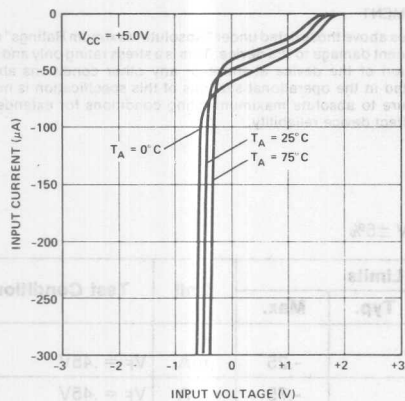
D.C. CHARACTERISTICS  $T_A = 0^\circ\text{C}$  to  $+75^\circ\text{C}$ ,  $V_{CC} = +5V \pm 5\%$ 

Symbol	Parameter	Limits			Unit	Test Conditions
		Min.	Typ.	Max.		
$I_F$	Input Load Current, ACK, DS <sub>2</sub> , CR, DI <sub>1</sub> -DI <sub>8</sub> Inputs			-.25	mA	$V_F = .45V$
$I_F$	Input Load Current MD Input			-.75	mA	$V_F = .45V$
$I_F$	Input Load Current DS <sub>1</sub> Input			-1.0	mA	$V_F = .45V$
$I_R$	Input Leakage Current, ACK, DS, CR, DI <sub>1</sub> -DI <sub>8</sub> Inputs			10	$\mu\text{A}$	$V_R \leq V_{CC}$
$I_R$	Input Leakage Current MO Input			30	$\mu\text{A}$	$V_R \leq V_{CC}$
$I_R$	Input Leakage Current DS <sub>1</sub> Input			40	$\mu\text{A}$	$V_R \leq V_{CC}$
$V_C$	Input Forward Voltage Clamp			-1	V	$I_C = -5\text{mA}$
$V_{IL}$	Input "Low" Voltage			.85	V	
$V_{IH}$	Input "High" Voltage	2.0			V	
$V_{OL}$	Output "Low" Voltage			.45	V	$I_{OL} = 15\text{mA}$
$V_{OH}$	Output "High" Voltage	3.65	4.0		V	$I_{OH} = -1\text{mA}$
$I_{SC}$	Short Circuit Output Current	-15		-75	mA	$V_O = 0V$ , $V_{CC} = 5V$
$ I_O $	Output Leakage Current High Impedance State			20	$\mu\text{A}$	$V_O = .45V/5.25V$
$I_{CC}$	Power Supply Current		90	130	mA	

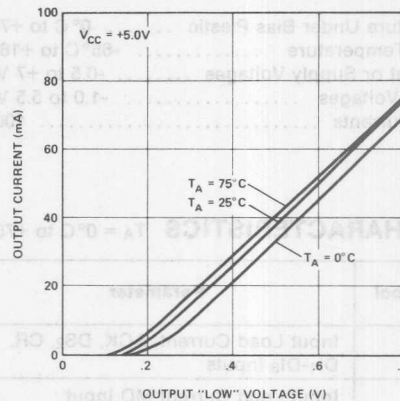


## TYPICAL CHARACTERISTICS

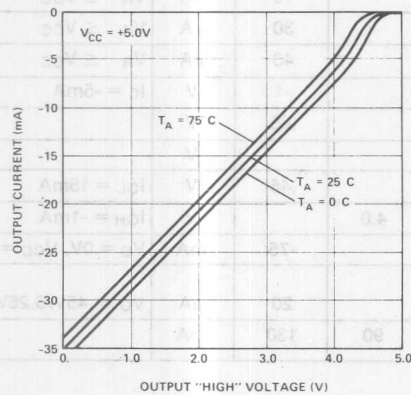
INPUT CURRENT VS. INPUT VOLTAGE



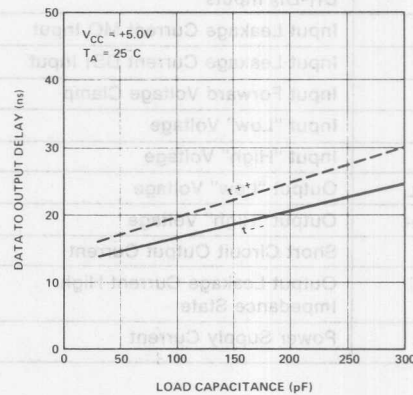
OUTPUT CURRENT VS. OUTPUT "LOW" VOLTAGE



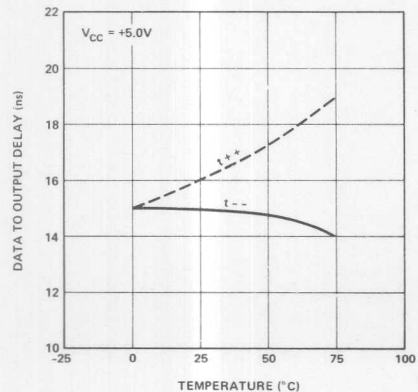
OUTPUT CURRENT VS. OUTPUT "HIGH" VOLTAGE



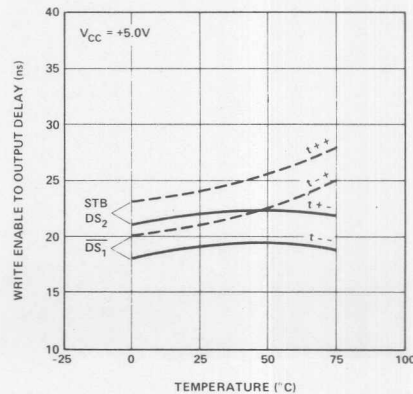
DATA TO OUTPUT DELAY VS. LOAD CAPACITANCE



DATA TO OUTPUT DELAY VS. TEMPERATURE



WRITE ENABLE TO OUTPUT DELAY VS. TEMPERATURE



**A.C. CHARACTERISTICS**  $T_A = 0^\circ\text{C}$  to  $+70^\circ\text{C}$ ,  $V_{CC} = +5\text{V} \pm 5\%$ 

Symbol	Parameter	Limits			Unit	Test Conditions
		Min.	Typ.	Max.		
tpw	Pulse Width	30			ns	
tpd	Data to Output Delay			30	ns	Note 1
twe	Write Enable to Output Delay			40	ns	Note 1
tset	Data Set Up Time	15			ns	
th	Data Hold Time	20			ns	
tr	Reset to Output Delay			40	ns	Note 1
ts	Set to Output Delay			30	ns	Note 1
te	Output Enable/Disable Time			45	ns	Note 1
tc	Clear to Output Delay			55	ns	Note 1

**CAPACITANCE\***  $F = 1\text{MHz}$ ,  $V_{BIAS} = 2.5\text{V}$ ,  $V_{CC} = +5\text{V}$ ,  $T_A = 25^\circ\text{C}$ 

Symbol	Test	Limits
		Typ. Max.
C <sub>IN</sub>	DS <sub>1</sub> MD Input Capacitance	9pF 12pF
C <sub>IN</sub>	DS <sub>2</sub> , CK, ACK, DI <sub>1</sub> -DI <sub>8</sub> Input Capacitance	5pF 9pF
C <sub>OUT</sub>	DO <sub>1</sub> -DO <sub>8</sub> Output Capacitance	8pF 12pF

\*This parameter is sampled and not 100% tested.

**SWITCHING CHARACTERISTICS****Conditions of Test**

Input Pulse Amplitude = 2.5V

Input Rise and Fall Times 5ns

Between 1V and 2V Measurements made at 1.5V with 15mA and 30pF Test Load

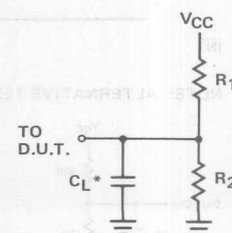
Note 1:

Test	C <sub>L</sub> *	R <sub>1</sub>	R <sub>2</sub>
tpd, twe, tr, ts, tc	30pF	300Ω	600Ω
te, ENABLE↑	30pF	10KΩ	1KΩ
te, ENABLE↓	30pF	300Ω	600Ω
te, DISABLE↑	5pF	300Ω	600Ω
te, DISABLE↓	5pF	10KΩ	1KΩ

\*Includes probe and jig capacitance.

**Test Load**

15mA & 30pF



\*INCLUDING JIG & PROBE CAPACITANCE



# 8257/8257-5

## PROGRAMMABLE DMA CONTROLLER

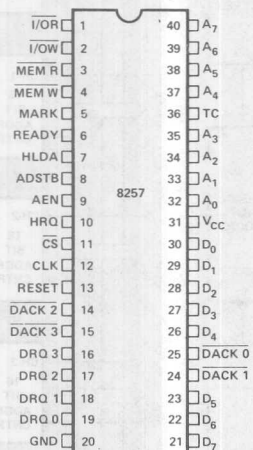
**PRELIMINARY**  
Notice: This is not a final specification. Some parametric limits are subject to change.

- MCS-85™ Compatible 8257-5
- 4-Channel DMA Controller
- Priority DMA Request Logic
- Channel Inhibit Logic

- Terminal Count and Modulo 128 Outputs
- Single TTL Clock
- Single +5V Supply
- Auto Load Mode

The Intel® 8257 is a 4-channel direct memory access (DMA) controller. It is specifically designed to simplify the transfer of data at high speeds for the Intel® microcomputer systems. Its primary function is to generate, upon a peripheral request, a sequential memory address which will allow the peripheral to read or write data directly to or from memory. Acquisition of the system bus in accomplished via the CPU's hold function. The 8257 has priority logic that resolves the peripherals requests and issues a composite hold request to the CPU. It maintains the DMA cycle count for each channel and outputs a control signal to notify the peripheral that the programmed number of DMA cycles is complete. Other output control signals simplify sectored data transfers. The 8257 represents a significant savings in component count for DMA-based microcomputer systems and greatly simplifies the transfer of data at high speed between peripherals and memories.

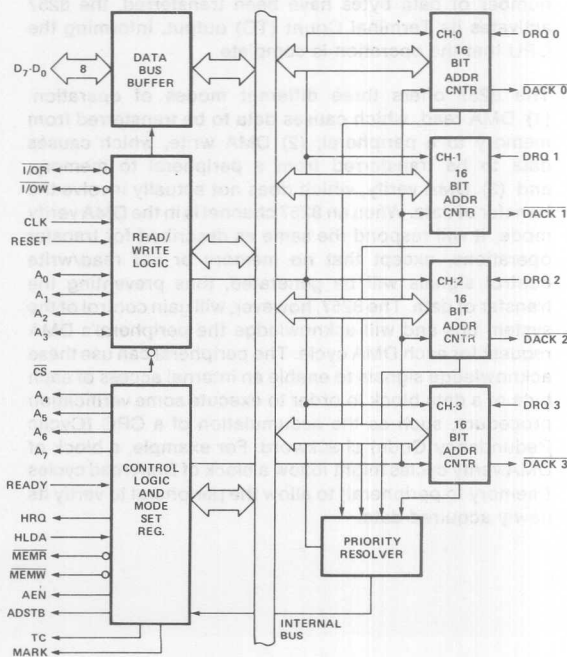
### PIN CONFIGURATION



### PIN NAMES

D7-D0	DATA BUS	AEN	ADDRESS ENABLE
A7-A0	ADDRESS BUS	ADSTB	ADDRESS STROBE
I/OR	I/O READ	TC	TERMINAL COUNT
I/OW	I/O WRITE	MARK	MODULO 128 MARK
MEMR	MEMORY READ	DRQ3-DRQ0	DMA REQUEST INPUT
MEMW	MEMORY WRITE	DACK3-DACK0	DMA ACKNOWLEDGE OUT
CLK	CLOCK INPUT	CS	CHIP SELECT
RESET	RESET INPUT	VCC	+5 VOLTS
READY	READY	GND	GROUND
HRQ	HOLD REQUEST (TO 8080A)		
HLDA	HOLD ACKNOWLEDGE (FROM 8080A)		

### BLOCK DIAGRAM



## FUNCTIONAL DESCRIPTION

### General

The 8257 is a programmable, Direct Memory Access (DMA) device which, when coupled with a single Intel® 8212 I/O port device, provides a complete four-channel DMA controller for use in Intel® microcomputer systems. After being initialized by software, the 8257 can transfer a block of data, containing up to 16,384 bytes, between memory and a peripheral device directly, without further intervention required of the CPU. Upon receiving a DMA transfer request from an enabled peripheral, the 8257:

1. Acquires control of the system bus.
2. Acknowledges that requesting peripheral which is connected to the highest priority channel.
3. Outputs the least significant eight bits of the memory address onto system address lines  $A_0$ - $A_7$ , outputs the most significant eight bits of the memory address to the 8212 I/O port via the data bus (the 8212 places these address bits on lines  $A_8$ - $A_{15}$ ), and
4. Generates the appropriate memory and I/O read/write control signals that cause the peripheral to receive or deposit a data byte directly from or to the addressed location in memory.

The 8257 will retain control of the system bus and repeat the transfer sequence, as long as a peripheral maintains its DMA request. Thus, the 8257 can transfer a block of data to/from a high speed peripheral (e.g., a sector of data on a floppy disk) in a single "burst". When the specified number of data bytes have been transferred, the 8257 activates its Terminal Count (TC) output, informing the CPU that the operation is complete.

The 8257 offers three different modes of operation: (1) DMA read, which causes data to be transferred from memory to a peripheral; (2) DMA write, which causes data to be transferred from a peripheral to memory; and (3) DMA verify, which does not actually involve the transfer of data. When an 8257 channel is in the DMA verify mode, it will respond the same as described for transfer operations, except that no memory or I/O read/write control signals will be generated, thus preventing the transfer of data. The 8257, however, will gain control of the system bus and will acknowledge the peripheral's DMA request for each DMA cycle. The peripheral can use these acknowledge signals to enable an internal access of each byte of a data block in order to execute some verification procedure, such as the accumulation of a CRC (Cyclic Redundancy Code) checkword. For example, a block of DMA verify cycles might follow a block of DMA read cycles (memory to peripheral) to allow the peripheral to verify its newly acquired data.

### Block Diagram Description

#### 1. DMA Channels

The 8257 provides four separate DMA channels (labeled CH-0 to CH-3). Each channel includes two sixteen-bit registers: (1) a DMA address register, and (2) a terminal count register. Both registers must be initialized before a channel is enabled. The DMA address register is loaded with the address of the first memory location to be accessed. The value loaded into the low-order 14-bits of the terminal count register specifies the number of DMA cycles minus one before the Terminal Count (TC) output is activated. For instance, a terminal count of 0 would cause the TC output to be active in the first DMA cycle for that channel. In general, if  $N$  = the number of desired DMA cycles, load the value  $N-1$  into the low-order 14-bits of the terminal count register. The most significant two bits of the terminal count register specify the type of DMA operation for that channel.

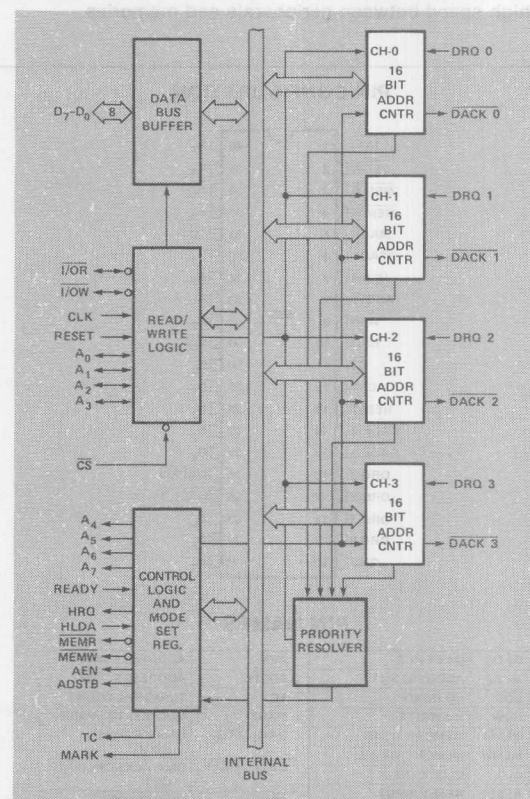


Figure 1. 8257 Block Diagram Showing DMA Channels

These two bits are not modified during a DMA cycle, but can be changed between DMA blocks.

Each channel accepts a DMA Request (DRQn) input and provides a DMA Acknowledge (DACKn) output.

### (DRQ 0-DRQ 3)

**DMA Request:** These are individual asynchronous channel request inputs used by the peripherals to obtain a DMA cycle. If not in the rotating priority mode then DRQ 0 has the highest priority and DRQ 3 has the lowest. A request can be generated by raising the request line and holding it high until DMA acknowledge. For multiple DMA cycles (Burst Mode) the request line is held high until the DMA acknowledge of the last cycle arrives.

### (DACK 0 - DACK 3)

**DMA Acknowledge:** An active low level on the acknowledge output informs the peripheral connected to that channel that it has been selected for a DMA cycle. The DACK output acts as a "chip select" for the peripheral device requesting service. This line goes active (low) and inactive (high) once for each byte transferred even if a burst of data is being transferred.

## 2. Data Bus Buffer

This three-state, bi-directional, eight bit buffer interfaces the 8257 to the system data bus.

### (D<sub>0</sub>-D<sub>7</sub>)

**Data Bus Lines:** These are bi-directional three-state lines. When the 8257 is being programmed by the CPU, eight-bits of data for a DMA address register, a terminal count register or the Mode Set register are received on the data bus. When the CPU reads a DMA address register, a terminal count register or the Status register, the data is sent to the CPU over the data bus. During DMA cycles (when the 8257 is the bus master), the 8257 will output the most significant eight-bits of the memory address (from one of the DMA address registers) to the 8212 latch via the data bus. These address bits will be transferred at the beginning of the DMA cycle; the bus will then be released to handle the memory data transfer during the balance of the DMA cycle.

BIT 15	BIT 14	TYPE OF DMA OPERATION
0	0	Verify DMA Cycle
0	1	Write DMA Cycle
1	0	Read DMA Cycle
1	1	(Illegal)

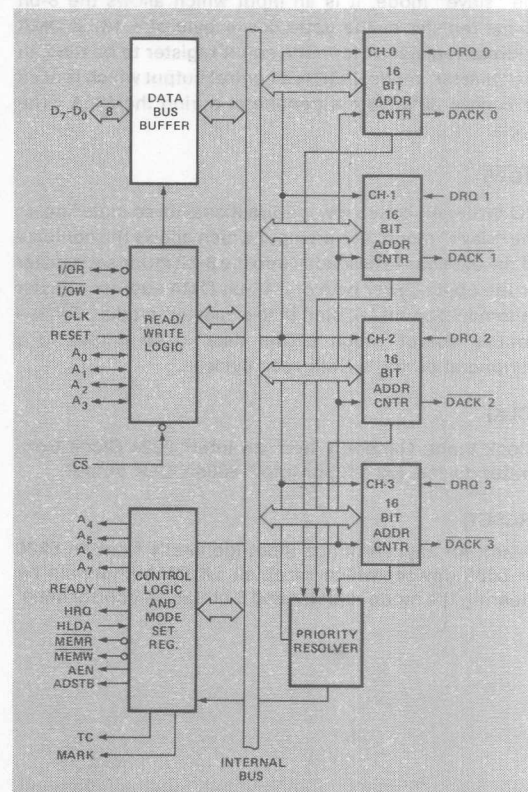


Figure 2. 8257 Block Diagram Showing Data Bus Buffer

### 3. Read/Write Logic

When the CPU is programming or reading one of the 8257's registers (i.e., when the 8257 is a "slave" device on the system bus), the Read/Write Logic accepts the I/O Read ( $\overline{I/OR}$ ) or I/O Write ( $\overline{I/OW}$ ) signal, decodes the least significant four address bits, ( $A_0-A_3$ ), and either writes the contents of the data bus into the addressed register (if  $\overline{I/OW}$  is true) or places the contents of the addressed register onto the data bus (if  $\overline{I/OR}$  is true).

During DMA cycles (i.e., when the 8257 is the bus "master"), the Read/Write Logic generates the I/O read and memory write (DMA write cycle) or I/O Write and memory read (DMA read cycle) signals which control the data link with the peripheral that has been granted the DMA cycle.

Note that during DMA transfers Non-DMA I/O devices should be de-selected (disabled) using "AEN" signal to inhibit I/O device decoding of the memory address as an erroneous device address.

#### $\overline{I/OR}$

I/O Read: An active-low, bi-directional three-state line. In the "slave" mode, it is an input which allows the 8-bit status register or the upper/lower byte of a 16-bit DMA address register or terminal count register to be read. In the "master" mode,  $\overline{I/OR}$  is a control output which is used to access data from a peripheral during the DMA write cycle.

#### $\overline{I/OW}$

I/O Write: An active-low, bi-directional three-state line. In the "slave" mode, it is an input which allows the contents of the data bus to be loaded into the 8-bit mode set register or the upper/lower byte of a 16-bit DMA address register or terminal count register. In the "master" mode,  $\overline{I/OW}$  is a control output which allows data to be output to a peripheral during a DMA read cycle.

#### (CLK)

Clock Input: Generally from an Intel® 8224 Clock Generator device. ( $\phi 2$  TTL) or Intel® 8085A CLK output.

#### (RESET)

Reset: An asynchronous input (generally from an 8224 or 8085 device) which disables all DMA channels by clearing the mode register and 3-states all control lines.

#### ( $A_0-A_3$ )

Address Lines: These least significant four address lines are bi-directional. In the "slave" mode they are inputs which select one of the registers to be read or programmed. In the "master" mode, they are outputs which constitute the least significant four bits of the 16-bit memory address generated by the 8257.

#### (CS)

Chip Select: An active-low input which enables the I/O Read or I/O Write input when the 8257 is being read or programmed in the "slave" mode. In the "master" mode, CS is automatically disabled to prevent the chip from selecting itself while performing the DMA function.

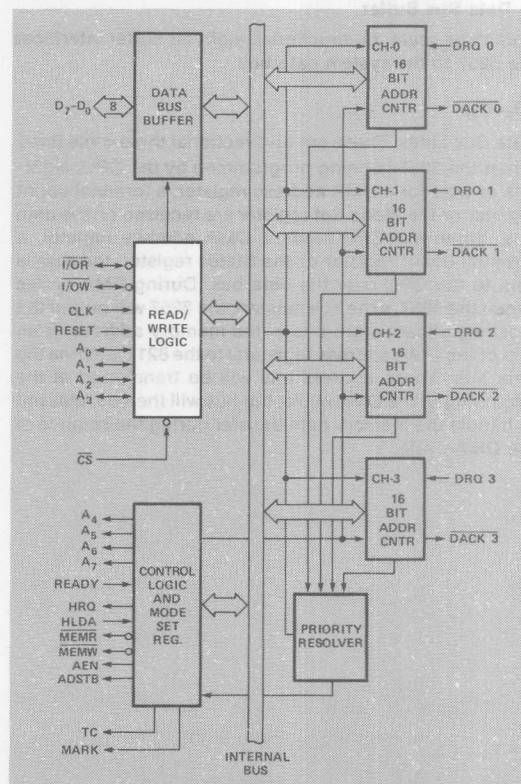


Figure 3. 8257 Block Diagram Showing Read/Write Logic Function

#### 4. Control Logic

This block controls the sequence of operations during all DMA cycles by generating the appropriate control signals and the 16-bit address that specifies the memory location to be accessed.

##### (A<sub>4</sub>-A<sub>7</sub>)

**Address Lines:** These four address lines are three-state outputs which constitute bits 4 through 7 of the 16-bit memory address generated by the 8257 during all DMA cycles.

##### (READY)

**Ready:** This asynchronous input is used to elongate the memory read and write cycles in the 8257 with wait states if the selected memory requires longer cycles.

##### (HRQ)

**Hold Request:** This output requests control of the system bus. In systems with only one 8257, HRQ will normally be applied to the HOLD input on the CPU.

##### (HLDA)

**Hold Acknowledge:** This input from the CPU indicates that the 8257 has acquired control of the system bus.

##### (MEMR)

**Memory Read:** This active-low three-state output is used to read data from the addressed memory location during DMA Read cycles.

##### (MEMW)

**Memory Write:** This active-low three-state output is used to write data into the addressed memory location during DMA Write cycles.

##### (ADSTB)

**Address Strobe:** This output strobes the most significant byte of the memory address into the 8212 device from the data bus.

##### (AEN)

**Address Enable:** This output is used to disable (float) the System Data Bus and the System Control Bus. It may also be used to disable (float) the System Address Bus by use of an enable on the Address Bus drivers in systems to inhibit non-DMA devices from responding during DMA cycles. It may be further used to isolate the 8257 data bus from the System Data Bus to facilitate the transfer of the 8 most significant DMA address bits over the 8257 data I/O pins without subjecting the System Data Bus to any timing constraints for the transfer. When the 8257 is used in an I/O device structure (as opposed to memory mapped), this AEN output should be used to disable the selection of an I/O device when the DMA address is on the address bus. The I/O device selection should be determined by the DMA acknowledge outputs for the 4 channels.

##### (TC)

**Terminal Count:** This output notifies the currently selected peripheral that the present DMA cycle should be the last cycle for this data block. If the TC STOP bit in the Mode Set register is set, the selected channel will be automatically disabled at the end of that DMA cycle. TC is activated when the 14-bit value in the selected channel's terminal count register equals zero. Recall that the low-order 14-bits of the terminal count register should be loaded with the values (n-1), where n = the desired number of the DMA cycles.

##### (MARK)

**Modulo 128 Mark:** This output notifies the selected peripheral that the current DMA cycle is the 128th cycle since the previous MARK output. MARK always occurs at 128 (and all multiples of 128) cycles from the end of the data block. Only if the total number of DMA cycles (n) is evenly divisible by 128 (and the terminal count register was loaded with n-1), will MARK occur at 128 (and each succeeding multiple of 128) cycles from the beginning of the data block.

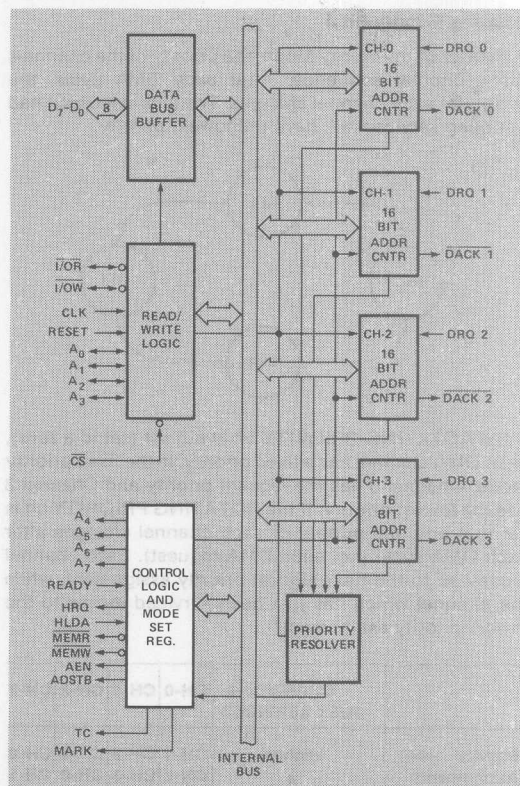
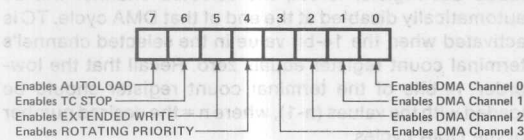


Figure 4. 8257 Block Diagram Showing Control Logic and Mode Set Register

## 5. Mode Set Register

When set, the various bits in the Mode Set register enable each of the four DMA channels, and allow four different options for the 8257:

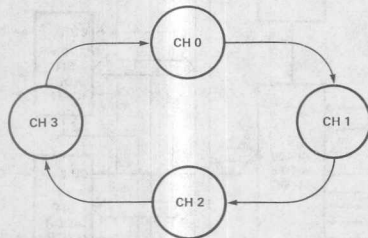


The Mode Set register is normally programmed by the CPU after the DMA address register(s) and terminal count register(s) are initialized. The Mode Set Register is cleared by the RESET input, thus disabling all options, inhibiting all channels, and preventing bus conflicts on power-up. A channel should not be left enabled unless its DMA address and terminal count registers contain valid values; otherwise, an inadvertent DMA request (DRQn) from a peripheral could initiate a DMA cycle that would destroy memory data.

The various options which can be enabled by bits in the Mode Set register are explained below:

### Rotating Priority Bit 4

In the Rotating Priority Mode, the priority of the channels has a circular sequence. After each DMA cycle, the priority of each channel changes. The channel which had just been serviced will have the lowest priority.



If the ROTATING PRIORITY bit is not set (set to a zero), each DMA channel has a fixed priority. In the fixed priority mode, Channel 0 has the highest priority and Channel 3 has the lowest priority. If the ROTATING PRIORITY bit is set to a one, the priority of each channel changes after each DMA cycle (not each DMA request). Each channel moves up to the next highest priority assignment, while the channel which has just been serviced moves to the lowest priority assignment:

	CHANNEL → JUST SERVICED	CH-0	CH-1	CH-2	CH-3
Priority → Assignments	Highest	CH-1	CH-2	CH-3	CH-0
	↑	CH-2	CH-3	CH-0	CH-1
	Lowest	CH-3	CH-0	CH-1	CH-2
		CH-0	CH-1	CH-2	CH-3

Note that rotating priority will prevent any one channel from monopolizing the DMA mode; consecutive DMA cycles will service different channels if more than one channel is enabled and requesting service. There is no overhead penalty associated with this mode of operation. All DMA operations began with Channel 0 initially assigned to the highest priority for the first DMA cycle.

### Extended Write Bit 5

If the EXTENDED WRITE bit is set, the duration of both the MEMW and I/OW signals is extended by activating them earlier in the DMA cycle. Data transfers within micro-computer systems proceed asynchronously to allow use of various types of memory and I/O devices with different access times. If a device cannot be accessed within a specific amount of time it returns a "not ready" indication to the 8257 that causes the 8257 to insert one or more wait states in its internal sequencing. Some devices are fast enough to be accessed without the use of wait states, but if they generate their READY response with the leading edge of the I/OW or MEMW signal (which generally occurs late in the transfer sequence), they would normally cause the 8257 to enter a wait state because it does not receive READY in time. For systems with these types of devices, the Extended Write option provides alternative timing for the I/O and memory write signals which allows the devices to return an early READY and prevents the unnecessary occurrence of wait states in the 8257, thus increasing system throughput.

### TC Stop Bit 6

If the TC STOP bit is set, a channel is disabled (i.e., its enable bit is reset) after the Terminal Count (TC) output goes true, thus automatically preventing further DMA operation on that channel. The enable bit for that channel must be re-programmed to continue or begin another DMA operation. If the TC STOP bit is not set, the occurrence of the TC output has no effect on the channel enable bits. In this case, it is generally the responsibility of the peripheral to cease DMA requests in order to terminate a DMA operation.

### Auto Load Bit 7

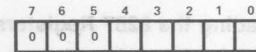
The Auto Load mode permits Channel 2 to be used for repeat block or block chaining operations, without immediate software intervention between blocks. Channel 2 registers are initialized as usual for the first data block; Channel 3 registers, however, are used to store the block re-initialization parameters (DMA starting address, terminal count and DMA transfer mode). After the first block of DMA cycles is executed by Channel 2 (i.e., after the TC output goes true), the parameters stored in the Channel 3 registers are transferred to Channel 2 during an "update" cycle. Note that the TC STOP feature, described above, has no effect on Channel 2 when the Auto Load bit is set.

If the Auto Load bit is set, the initial parameters for Channel 2 are automatically duplicated in the Channel 3 registers when Channel 2 is programmed. This permits repeat block operations to be set up with the programming of a single channel. Repeat block operations can be used in applications such as CRT refreshing. Channels 2 and 3 can still be loaded with separate values if Channel 2 is loaded before loading Channel 3. Note that in the Auto Load mode, Channel 3 is still available to the user if the Channel 3 enable bit is set, but use of this channel will change the values to be auto loaded into Channel 2 at update time. All that is necessary to use the Auto Load feature for chaining operations is to reload Channel 3 registers at the conclusion of each update cycle with the new parameters for the next data block transfer.

Each time that the 8257 enters an update cycle, the update flag in the status register is set and parameters in Channel 3 are transferred to Channel 2, non-destructively for Channel 3. The actual re-initialization of Channel 2 occurs at the beginning of the next channel 2 DMA cycle after the TC cycle. This will be the first DMA cycle of the new data block for Channel 2. The update flag is cleared at the conclusion of this DMA cycle. For chaining operations, the update flag in the status register can be monitored by the CPU to determine when the re-initialization process has been completed so that the next block parameters can be safely loaded into Channel 3.

## 6. Status Register

The eight-bit status register indicates which channels have reached a terminal count condition and includes the update flag described previously.



The TC status bits are set when the Terminal Count (TC) output is activated for that channel. These bits remain set until the status register is read or the 8257 is reset. The UPDATE FLAG, however, is not affected by a status register read operation. The UPDATE FLAG can be cleared by resetting the 8257, by changing to the non-auto load mode (i.e., by resetting the AUTO LOAD bit in the Mode Set register) or it can be left to clear itself at the completion of the update cycle. The purpose of the UPDATE FLAG is to prevent the CPU from inadvertently skipping a data block by overwriting a starting address or terminal count in the Channel 3 registers before those parameters are properly auto-loaded into Channel 2.

The user is cautioned against reading the TC status register and using this information to reenable channels that have not completed operation. Unless the DMA channels are inhibited a channel could reach terminal count (TC) between the status read and the mode write. DMA can be inhibited by a hardware gate on the HRQ line or by disabling channels with a mode word before reading the TC status.

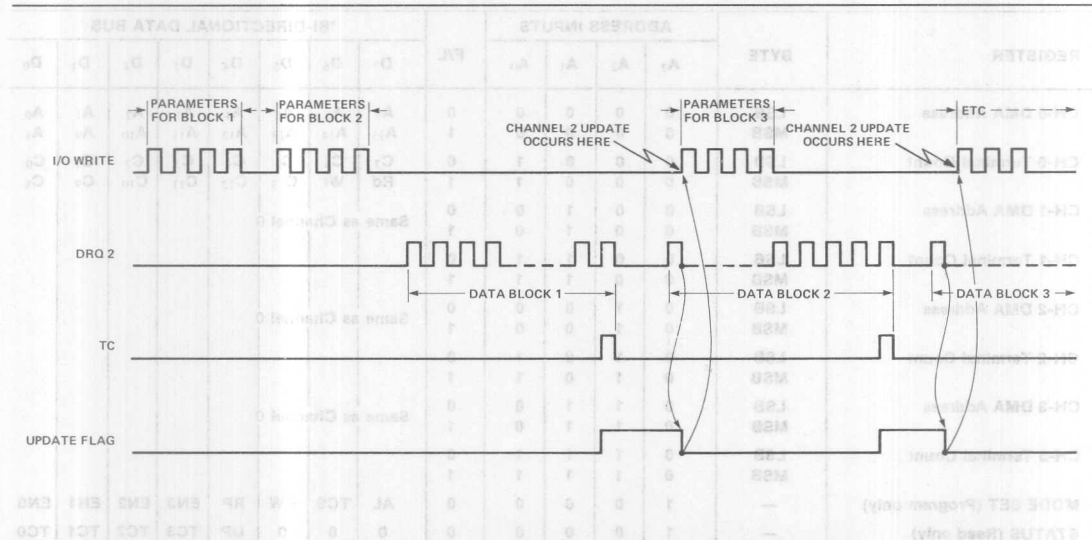


Figure 5. Autoload Timing

## OPERATIONAL SUMMARY

### Programming and Reading the 8257 Registers

There are four pairs of "channel registers": each pair consisting of a 16-bit DMA address register and a 16-bit terminal count register (one pair for each channel). The 8257 also includes two "general registers": one 8-bit Mode Set register and one 8-bit Status register. The registers are loaded or read when the CPU executes a write or read instruction that addresses the 8257 device and the appropriate register within the 8257. The 8228 generates the appropriate read or write control signal (generally I/OR or I/OW while the CPU places a 16-bit address on the system address bus, and either outputs the data to be written onto the system data bus or accepts the data being read from the data bus. All or some of the most significant 12 address bits A<sub>4</sub>-A<sub>15</sub> (depending on the systems memory, I/O configuration) are usually decoded to produce the chip select (CS) input to the 8257. An I/O Write input (or Memory Write in memory mapped I/O configurations, described below) specifies that the addressed register is to be programmed, while an I/O Read input (or Memory Read) specifies that the addressed register is to be read. Address bit 3 specifies whether a "channel register" (A<sub>3</sub> = 0) or the Mode Set (program only)/Status (read only) register (A<sub>3</sub> = 1) is to be accessed.

The least significant three address bits, A<sub>0</sub>-A<sub>2</sub>, indicate the specific register to be accessed. When accessing the Mode Set or Status register, A<sub>0</sub>-A<sub>2</sub> are all zero. When accessing a channel register bit A<sub>0</sub> differentiates between the DMA address register (A<sub>0</sub> = 0) and the terminal count register (A<sub>0</sub> = 1), while bits A<sub>1</sub> and A<sub>2</sub> specify one of the

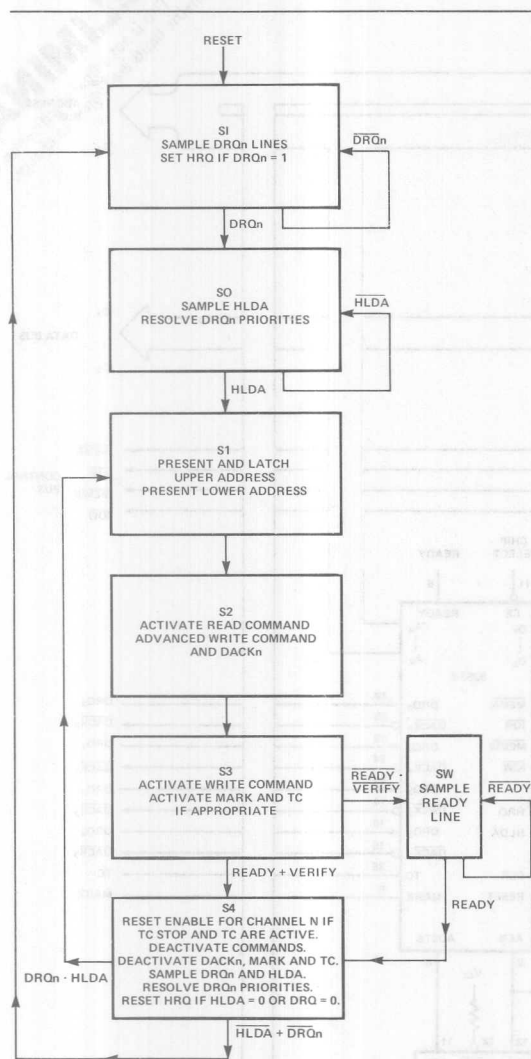
CONTROL INPUT	CS	I/OW	I/OR	A <sub>3</sub>
Program Half of a Channel Register	0	0	1	0
Read Half of a Channel Register	0	1	0	0
Program Mode Set Register	0	0	1	1
Read Status Register	0	1	0	1

four channels. Because the "channel registers" are 16-bits, two program instruction cycles are required to load or read an entire register. The 8257 contains a first/last (F/L) flip flop which toggles at the completion of each channel program or read operation. The F/L flip flop determines whether the upper or lower byte of the register is to be accessed. The F/L flip flop is reset by the RESET input and whenever the Mode Set register is loaded. To maintain proper synchronization when accessing the "channel registers" all channel command instruction operations should occur in pairs, with the lower byte of a register always being accessed first. Do not allow CS to clock while either I/OR or I/OW is active, as this will cause an erroneous F/L flip flop state. In systems utilizing an interrupt structure, interrupts should be disabled prior to any paired programming operations to prevent an interrupt from splitting them. The result of such a split would leave the F/L F/F in the wrong state. This problem is particularly obvious when other DMA channels are programmed by an interrupt structure.

### 8257 Register Selection

REGISTER	BYTE	ADDRESS INPUTS				F/L	*BI-DIRECTIONAL DATA BUS							
		A <sub>3</sub>	A <sub>2</sub>	A <sub>1</sub>	A <sub>0</sub>		D <sub>7</sub>	D <sub>6</sub>	D <sub>5</sub>	D <sub>4</sub>	D <sub>3</sub>	D <sub>2</sub>	D <sub>1</sub>	D <sub>0</sub>
CH-0 DMA Address	LSB	0	0	0	0	0	A <sub>7</sub>	A <sub>6</sub>	A <sub>5</sub>	A <sub>4</sub>	A <sub>3</sub>	A <sub>2</sub>	A <sub>1</sub>	A <sub>0</sub>
	MSB	0	0	0	0	1	A <sub>15</sub>	A <sub>14</sub>	A <sub>13</sub>	A <sub>12</sub>	A <sub>11</sub>	A <sub>10</sub>	A <sub>9</sub>	A <sub>8</sub>
CH-0 Terminal Count	LSB	0	0	0	1	0	C <sub>7</sub>	C <sub>6</sub>	C <sub>5</sub>	C <sub>4</sub>	C <sub>3</sub>	C <sub>2</sub>	C <sub>1</sub>	C <sub>0</sub>
	MSB	0	0	0	1	1	Rd	Wr	C <sub>13</sub>	C <sub>12</sub>	C <sub>11</sub>	C <sub>10</sub>	C <sub>9</sub>	C <sub>8</sub>
CH-1 DMA Address	LSB	0	0	1	0	0	Same as Channel 0							
	MSB	0	0	1	0	1								
CH-1 Terminal Count	LSB	0	0	1	1	0	Same as Channel 0							
	MSB	0	0	1	1	1								
CH-2 DMA Address	LSB	0	1	0	0	0	Same as Channel 0							
	MSB	0	1	0	0	1								
CH-2 Terminal Count	LSB	0	1	0	1	0	Same as Channel 0							
	MSB	0	1	0	1	1								
CH-3 DMA Address	LSB	0	1	1	0	0	Same as Channel 0							
	MSB	0	1	1	0	1								
CH-3 Terminal Count	LSB	0	1	1	1	0								
	MSB	0	1	1	1	1								
MODE SET (Program only)	—	1	0	0	0	0	AL	TCS	EW	RP	EN3	EN2	EN1	EN0
STATUS (Read only)	—	1	0	0	0	0	0	0	0	UP	TC3	TC2	TC1	TC0

\*A<sub>0</sub>-A<sub>15</sub>: DMA Starting Address, C<sub>0</sub>-C<sub>13</sub>: Terminal Count value (N-1), Rd and Wr: DMA Verify (00), Write (01) or Read (10) cycle selection, AL: Auto Load, TCS: TC STOP, EW: EXTENDED WRITE, RP: ROTATING PRIORITY, EN3-EN0: CHANNEL ENABLE MASK, UP: UPDATE FLAG, TC3-TC0: TERMINAL COUNT STATUS BITS.



1 DRQn refers to any DRQ line on an enabled DMA channel.

Figure 6. DMA Operation State Diagram

## DMA OPERATION

### Single Byte Transfers

A single byte transfer is initiated by the I/O device raising the DRQ line of one channel of the 8257. If the channel is enabled, the 8257 will output a HRQ to the CPU. The 8257 now waits until a HLDA is received insuring that the system bus is free for its use. Once HLDA is received the DACK line for the requesting channel is activated (LOW). The DACK line acts as a chip select for the requesting I/O device. The 8257 then generates the

read and write commands and byte transfer occurs between the selected I/O device and memory. After the transfer is complete, the DACK line is set HIGH and the HRQ line is set LOW to indicate to the CPU that the bus is now free for use. DRQ must remain HIGH until DACK is issued to be recognized and must go LOW before S4 of the transfer sequence to prevent another transfer from occurring. (See timing diagram.)

### Consecutive Transfers

If more than one channel requests service simultaneously, the transfer will occur in the same way a burst does. No overhead is incurred by switching from one channel to another. In each S4 the DRQ lines are sampled and the highest priority request is recognized during the next transfer. A burst mode transfer in a lower priority channel will be overridden by a higher priority request. Once the high priority transfer has completed control will return to the lower priority channel if its DRQ is still active. No extra cycles are needed to execute this sequence and the HRQ line remains active until all DRQ lines go LOW.

### Control Override

The continuous DMA transfer mode described above can be interrupted by an external device by lowering the HLDA line. After each DMA transfer the 8257 samples the HLDA line to insure that it is still active. If it is not active, the 8257 completes the current transfer, releases the HRQ line (LOW) and returns to the idle state. If DRQ lines are still active the 8257 will raise the HRQ line in the third cycle and proceed normally. (See timing diagram.)

### Not Ready

The 8257 has a Ready input similar to the 8080A and the 8085A. The Ready line is sampled in State 3. If Ready is LOW the 8257 enters a wait state. Ready is sampled during every wait state. When Ready returns HIGH the 8257 proceeds to State 4 to complete the transfer. Ready is used to interface memory or I/O devices that cannot meet the bus set up times required by the 8257.

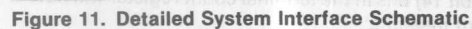
### Speed

The 8257 uses four clock cycles to transfer a byte of data. No cycles are lost in the master to master transfer maximizing bus efficiency. A 2MHz clock input will allow the 8257 to transfer at a rate of 500K bytes/second.

### Memory Mapped I/O Configurations

The 8257 can be connected to the system bus as a memory device instead of as an I/O device for memory mapped I/O configurations by connecting the system memory control lines to the 8257's I/O control lines and the system I/O control lines to the 8257's memory control lines.

This configuration permits use of the 8080's considerably larger repertoire of memory instructions when reading or loading the 8257's registers. Note that with this connection, the programming of the Read (bit 15) and Write (bit 14) bits in the terminal count register will have a different meaning:



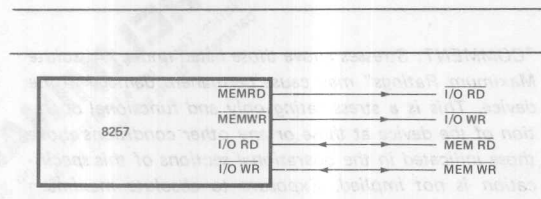


Figure 7. System Interface for Memory Mapped I/O

BIT 15 READ	BIT 14 WRITE	
0	0	DMA Verify Cycle
0	1	DMA Read Cycle
1	0	DMA Write Cycle
1	1	Illegal

Figure 8. TC Register for Memory Mapped I/O Only

## SYSTEM APPLICATION EXAMPLES

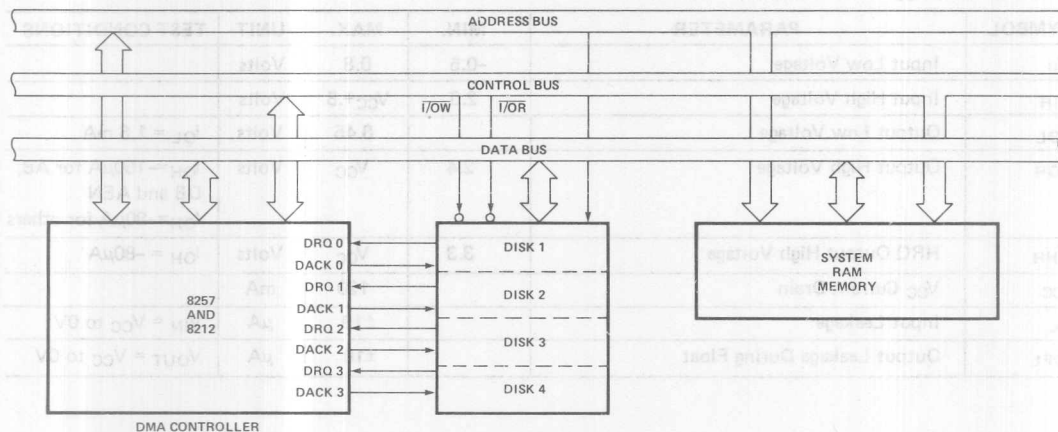


Figure 9. Floppy Disk Controller (4 Drives)

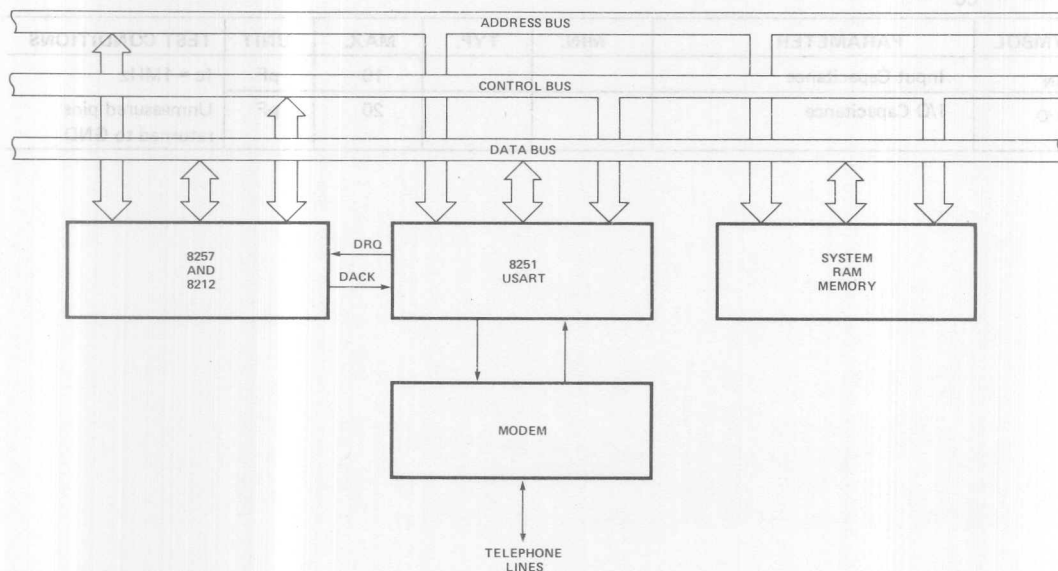


Figure 10. High-Speed Communication Controller

Ambient Temperature Under Bias. . . . . 0°C to 70°C  
Storage Temperature . . . . . -65°C to +150°C  
Voltage on Any Pin  
With Respect to Ground. . . . . -0.5V to +7V  
Power Dissipation . . . . . 1 Watt

device. This is a stress rating only and functional operation of the device at these or any other conditions above those indicated in the operational sections of this specification is not implied. Exposure to absolute maximum rating conditions for extended periods may affect device reliability.

## D.C. CHARACTERISTICS

T<sub>A</sub> = 0°C to 70°C, V<sub>CC</sub> = +5V ± 5%, GND = 0V

SYMBOL	PARAMETER	MIN.	MAX.	UNIT	TEST CONDITIONS
V <sub>IL</sub>	Input Low Voltage	-0.5	0.8	Volts	
V <sub>IH</sub>	Input High Voltage	2.0	V <sub>CC</sub> +1.5	Volts	
V <sub>OL</sub>	Output Low Voltage		0.45	Volts	I <sub>OL</sub> = 1.6 mA
V <sub>OH</sub>	Output High Voltage	2.4	V <sub>CC</sub>	Volts	I <sub>OH</sub> = -150μA for AB, DB and AEN I <sub>OH</sub> = -80μA for others
V <sub>HH</sub>	HRQ Output High Voltage	3.3	V <sub>CC</sub>	Volts	I <sub>OH</sub> = -80μA
I <sub>CC</sub>	V <sub>CC</sub> Current Drain		120	mA	
I <sub>IL</sub>	Input Leakage		±10	μA	V <sub>IN</sub> = V <sub>CC</sub> to 0V
I <sub>OL</sub>	Output Leakage During Float		±10	μA	V <sub>OUT</sub> = V <sub>CC</sub> to 0V

## CAPACITANCE

T<sub>A</sub> = 25°C; V<sub>CC</sub> = GND = 0V

SYMBOL	PARAMETER	MIN.	TYP.	MAX.	UNIT	TEST CONDITIONS
C <sub>IN</sub>	Input Capacitance			10	pF	f <sub>c</sub> = 1MHz
C <sub>I/O</sub>	I/O Capacitance			20	pF	Unmeasured pins returned to GND

**A.C. CHARACTERISTICS: PERIPHERAL (SLAVE) MODE**

$T_A = 0^\circ\text{C}$  to  $70^\circ\text{C}$ ,  $V_{CC} = 5.0\text{V} \pm 5\%$ ; GND = 0V (Note 1).

**8080 Bus Parameters****Read Cycle:**

Symbol	Parameter	8257		8257-5		Unit	Test Conditions
		Min.	Max.	Min.	Max.		
$T_{AR}$	Adr or $\overline{CS}\downarrow$ Setup to $\overline{RD}\downarrow$	0		0		ns	
$T_{RA}$	Adr or $\overline{CS}\uparrow$ Hold from $\overline{RD}\uparrow$	0		0		ns	
$T_{RD}$	Data Access from $\overline{RD}\downarrow$	0	300	0	200	ns	(Note 2)
$T_{DF}$	DB $\rightarrow$ Float Delay from $\overline{RD}\uparrow$	20	150	20	100	ns	
$T_{RR}$	$\overline{RD}$ Width	250		250		ns	

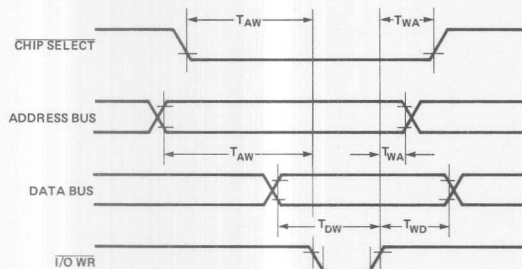
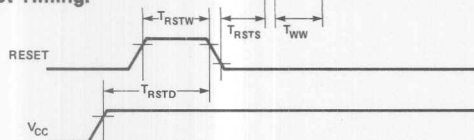
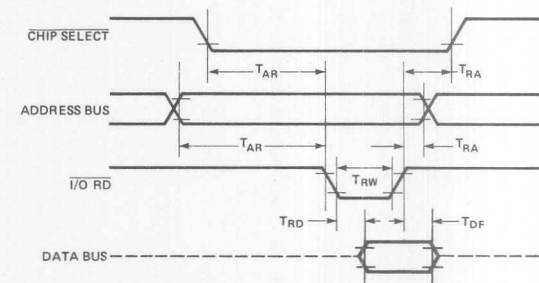
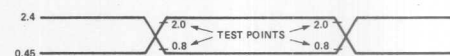
**Write Cycle:**

Symbol	Parameter	8257		8257-5		Unit	Test Conditions
		Min.	Max.	Min.	Max.		
$T_{AW}$	Adr Setup to $\overline{WR}\downarrow$	20		20		ns	
$T_{WA}$	Adr Hold from $\overline{WR}\uparrow$	0		0		ns	
$T_{DW}$	Data Setup to $\overline{WR}\uparrow$	200		200		ns	
$T_{WD}$	Data Hold from $\overline{WR}\uparrow$	0		0		ns	
$T_{WW}$	$\overline{WR}$ Width	200		200		ns	

**Other Timing:**

Symbol	Parameter	8257		8257-5		Unit	Test Conditions
		Min.	Max.	Min.	Max.		
$T_{RSTW}$	Reset Pulse Width	300		300		ns	
$T_{RSTD}$	Power Supply $\uparrow$ ( $V_{CC}$ ) Setup to Reset $\downarrow$	500		500		$\mu\text{s}$	
$T_r$	Signal Rise Time		20		20	ns	
$T_f$	Signal Fall Time		20		20	ns	
$T_{RSTS}$	Reset to First $\overline{I/O}\overline{WR}$	2		2		$t_{CY}$	

Notes: 1. All timing measurements are made at the following reference voltages unless specified otherwise: Input "1" at 2.0V, "0" at 0.8V  
 2. 8257:  $C_L = 100\text{pF}$ , 8257-5:  $C_L = 150\text{pF}$ . Output "1" at 2.0V, "0" at 0.8V

**8257 PERIPHERAL MODE TIMING DIAGRAMS****Write Timing:****Reset Timing:****Read Timing:****Input Waveform for A.C. Tests:**

**A.C. CHARACTERISTICS: DMA (MASTER) MODE**  $T_A = 0^\circ\text{C to } 70^\circ\text{C}$ ,  $V_{CC} = +5V \pm 5\%$ ,  $GND = 0V$ **Timing Requirements**

SYMBOL	PARAMETER	8257		8257-5		UNIT
		MIN.	MAX.	MIN.	MAX.	
$T_{CY}$	Cycle Time (Period)	0.320	4	0.320	4	$\mu\text{s}$
$T_\theta$	Clock Active (High)	120	$.8T_{CY}$	80	$.8T_{CY}$	ns
$T_{QS}$	$DRQ\uparrow$ Setup to $\theta\downarrow$ (S1, S4)	120		120		ns
$T_{QH}$	$DRQ\downarrow$ Hold from $HLDA\uparrow$ [4]	0		0		ns
$T_{HS}$	$HLDA\uparrow$ or $\downarrow$ Setup to $\theta\downarrow$ (S1, S4)	100		100		ns
$T_{RS}$	READY Setup Time to $\theta\uparrow$ (S3, Sw)	30		30		ns
$T_{RH}$	READY Hold Time from $\theta\uparrow$ (S3, Sw)	20		20		ns

Note: 4. Tracking Parameter.

**Tracking Parameters**

Signals labeled as Tracking Parameters (footnotes 4-7 under A.C. Specifications) are signals that follow similar paths through the silicon die. The propagation speed of these signals varies in the manufacturing process but the relationship between all these parameters is constant. The variation is less than or equal to 50 ns.

Suppose the following timing equation is being evaluated,

$$T_{A(\text{MIN})} + T_{B(\text{MAX})} \leq 150 \text{ ns}$$

and only minimum specifications exist for  $T_A$  and  $T_B$ . If  $T_{A(\text{MIN})}$  is used, and if  $T_A$  and  $T_B$  are tracking parameters,  $T_{B(\text{MAX})}$  can be taken as  $T_{B(\text{MIN})} + 50 \text{ ns}$ .

$$T_{A(\text{MIN})} + (T_{B(\text{MIN})} + 50 \text{ ns}) \leq 150 \text{ ns}$$

\*if  $T_A$  and  $T_B$  are tracking parameters

**A.C. CHARACTERISTICS: DMA (MASTER) MODE**  $T_A = 0^\circ\text{C to } 70^\circ\text{C}$ ,  $V_{CC} = +5V \pm 5\%$ ,  $GND = 0V$ **Timing Responses**

SYMBOL	PARAMETER	8257		8257-5		UNIT
		MIN.	MAX.	MIN.	MAX.	
$T_{DQ}$	HRQ $\uparrow$ or $\downarrow$ Delay from $\theta\uparrow$ (SI,S4) (measured at 2.0V) <sup>[1]</sup>		160		160	ns
$T_{DQ1}$	HRQ $\uparrow$ or $\downarrow$ Delay from $\theta\uparrow$ (SI,S4) (measured at 3.3V) <sup>[3]</sup>		250		250	ns
$T_{AEL}$	AEN $\uparrow$ Delay from $\theta\downarrow$ (S1) <sup>[1]</sup>		300		300	ns
$T_{AET}$	AEN $\downarrow$ Delay from $\theta\uparrow$ (S1) <sup>[1]</sup>		200		200	ns
$T_{AEA}$	Adr(AB)(Active) Delay from AEN $\uparrow$ (S1) <sup>[4]</sup>	20		20		ns
$T_{FAAB}$	Adr(AB)(Active) Delay from $\theta\uparrow$ (S1) <sup>[2]</sup>		250		250	ns
$T_{AFAB}$	Adr(AB)(Float) Delay from $\theta\uparrow$ (S1) <sup>[2]</sup>		150		150	ns
$T_{ASM}$	Adr(AB)(Stable) Delay from $\theta\uparrow$ (S1) <sup>[2]</sup>		250		250	ns
$T_{AH}$	Adr(AB)(Stable) Hold from $\theta\uparrow$ (S1) <sup>[2]</sup>	$T_{ASM}-50$		$T_{ASM}-50$		ns
$T_{AHR}$	Adr(AB)(Valid) Hold from $Rd\uparrow$ (S1,S1) <sup>[4]</sup>	60		60		ns
$T_{AHW}$	Adr(AB)(Valid) Hold from $Wr\uparrow$ (S1,S1) <sup>[4]</sup>	300		300		ns
$T_{FADB}$	Adr(DB)(Active) Delay from $\theta\uparrow$ (S1) <sup>[2]</sup>		300		300	ns
$T_{AFDB}$	Adr(DB)(Float) Delay from $\theta\uparrow$ (S2) <sup>[2]</sup>	$T_{STT}+20$	250	$T_{STT}+20$	170	ns
$T_{ASS}$	Adr(DB) Setup to AdrStb $\downarrow$ (S1-S2) <sup>[4]</sup>	100		100		ns
$T_{AHS}$	Adr(DB)(Valid) Hold from AdrStb $\downarrow$ (S2) <sup>[4]</sup>	50		50		ns
$T_{STL}$	AdrStb $\uparrow$ Delay from $\theta\uparrow$ (S1) <sup>[1]</sup>		200		200	ns
$T_{STT}$	AdrStb $\downarrow$ Delay from $\theta\uparrow$ (S2) <sup>[1]</sup>		140		140	ns
$T_{SW}$	AdrStb Width (S1-S2) <sup>[4]</sup>	$T_{CY}-100$		$T_{CY}-100$		ns
$T_{ASC}$	$Rd\downarrow$ or $Wr$ (Ext) $\downarrow$ Delay from AdrStb $\downarrow$ (S2) <sup>[4]</sup>	70		70		ns
$T_{DBC}$	$Rd\downarrow$ or $Wr$ (Ext) $\downarrow$ Delay from Adr(DB) (Float)(S2) <sup>[4]</sup>	20		20		ns
$T_{AK}$	DACK $\uparrow$ or $\downarrow$ Delay from $\theta\downarrow$ (S2,S1) and TC/Mark $\uparrow$ Delay from $\theta\uparrow$ (S3) and TC/Mark $\downarrow$ Delay from $\theta\uparrow$ (S4) <sup>[1,5]</sup>		250		250	ns
$T_{DCL}$	$Rd\downarrow$ or $Wr$ (Ext) $\downarrow$ Delay from $\theta\uparrow$ (S2) and $Wr\downarrow$ Delay from $\theta\uparrow$ (S3) <sup>[2,6]</sup>		200		200	ns
$T_{DCT}$	$Rd\uparrow$ Delay from $\theta\downarrow$ (S1,S1) and $Wr\uparrow$ Delay from $\theta\uparrow$ (S4) <sup>[2,7]</sup>		200		200	ns
$T_{FAC}$	$Rd$ or $Wr$ (Active) from $\theta\uparrow$ (S1) <sup>[2]</sup>		300		300	ns
$T_{AFC}$	$Rd$ or $Wr$ (Float) from $\theta\uparrow$ (S1) <sup>[2]</sup>		150		150	ns
$T_{RWM}$	$Rd$ Width (S2-S1 or S1) <sup>[4]</sup>	$2T_{CY} + T_\theta - 50$		$2T_{CY} + T_\theta - 50$		ns
$T_{WWM}$	$Wr$ Width (S3-S4) <sup>[4]</sup>	$T_{CY}-50$		$T_{CY}-50$		ns
$T_{WWME}$	$Wr$ (Ext) Width (S2-S4) <sup>[4]</sup>	$2T_{CY}-50$		$2T_{CY}-50$		ns

Notes: 1. Load = 1 TTL. 2. Load = 1 TTL + 50pF. 3. Load = 1 TTL + ( $R_L = 3.3K$ ),  $V_{OH} = 3.3V$ . 4. Tracking Parameter.  
5.  $\Delta T_{AK} < 50$  ns. 6.  $\Delta T_{DCL} < 50$  ns. 7.  $\Delta T_{DCT} < 50$  ns.

[illegible]

5-84

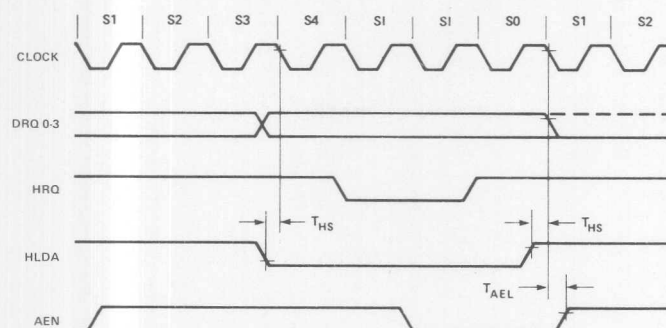


Figure 13. Control Override Sequence

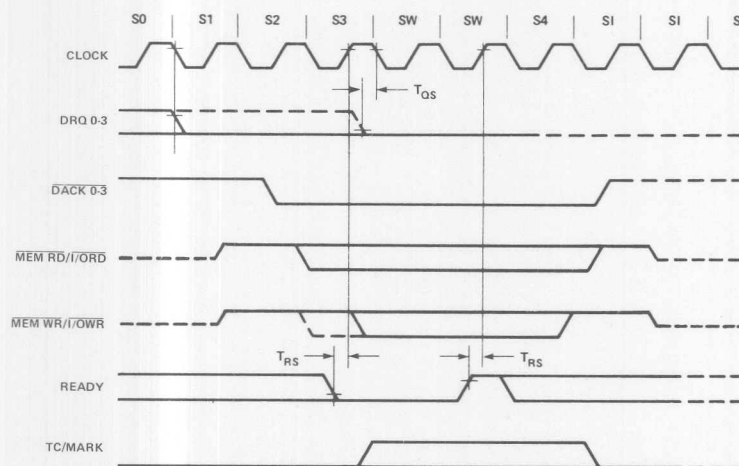


Figure 14. Not Ready Sequence



# 8259A PROGRAMMABLE INTERRUPT CONTROLLER

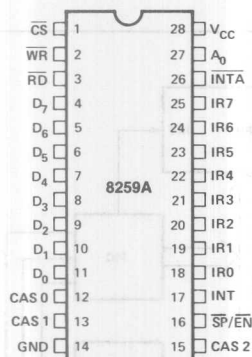
- MCS-86™ Compatible
- MCS-80/85™ Compatible
- Eight-Level Priority Controller
- Expandable to 64 Levels
- Programmable Interrupt Modes
- Individual Request Mask Capability
- Single +5V Supply (No Clocks)
- 28-Pin Dual-In-Line Package

The Intel® 8259A Programmable Interrupt Controller handles up to eight vectored priority interrupts for the CPU. It is cascadable for up to 64 vectored priority interrupts without additional circuitry. It is packaged in a 28-pin DIP, uses NMOS technology and requires a single +5V supply. Circuitry is static, requiring no clock input.

The 8259A is designed to minimize the software and real time overhead in handling multi-level priority interrupts. It has several modes, permitting optimization for a variety of system requirements.

The 8259A is fully upward compatible with the Intel® 8259. Software originally written for the 8259 will operate the 8259A in all 8259 equivalent modes (MCS-80/85, Non-Buffered, Edge Triggered).

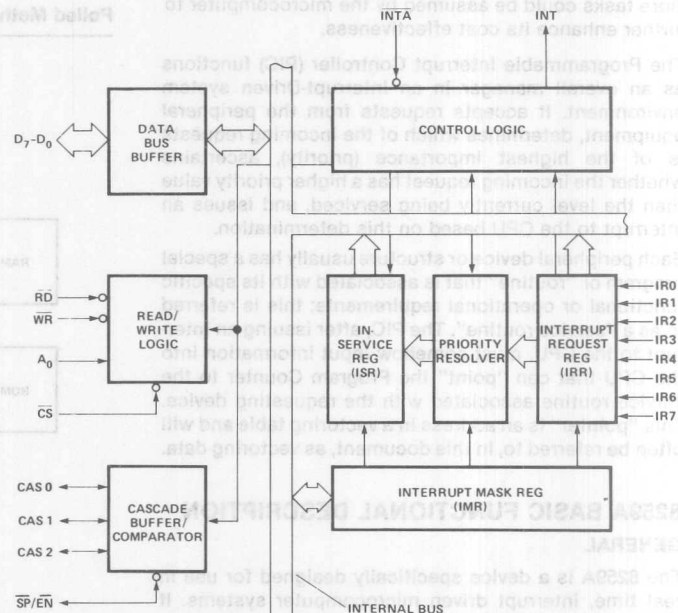
## PIN CONFIGURATION



## PIN NAMES

D <sub>7</sub> -D <sub>0</sub>	DATA BUS (BI-DIRECTIONAL)
RD	READ INPUT
WR	WRITE INPUT
A <sub>0</sub>	COMMAND SELECT ADDRESS
CS	CHIP SELECT
CAS2-CAS0	CASCADE LINES
SP/EN	SLAVE PROGRAM INPUT/ENABLE
INT	INTERRUPT OUTPUT
INTA	INTERRUPT ACKNOWLEDGE INPUT
IR0-IR7	INTERRUPT REQUEST INPUTS

## BLOCK DIAGRAM



## INTERRUPTS IN MICROCOMPUTER SYSTEMS

Microcomputer system design requires that I/O devices such as keyboards, displays, sensors and other components receive servicing in an efficient manner so that large amounts of the total system tasks can be assumed by the microcomputer with little or no effect on throughput.

The most common method of servicing such devices is the *Polled* approach. This is where the processor must test each device in sequence and in effect "ask" each one if it needs servicing. It is easy to see that a large portion of the main program is looping through this continuous polling cycle and that such a method would have a serious, detrimental effect on system throughput, thus limiting the tasks that could be assumed by the microcomputer and reducing the cost effectiveness of using such devices.

A more desirable method would be one that would allow the microprocessor to be executing its main program and only stop to service peripheral devices when it is told to do so by the device itself. In effect, the method would provide an external asynchronous input that would inform the processor that it should complete whatever instruction that is currently being executed and fetch a new routine that will service the requesting device. Once this servicing is complete, however, the processor would resume exactly where it left off.

This method is called *Interrupt*. It is easy to see that system throughput would drastically increase, and thus more tasks could be assumed by the microcomputer to further enhance its cost effectiveness.

The Programmable Interrupt Controller (PIC) functions as an overall manager in an Interrupt-Driven system environment. It accepts requests from the peripheral equipment, determines which of the incoming requests is of the highest importance (priority), ascertains whether the incoming request has a higher priority value than the level currently being serviced, and issues an interrupt to the CPU based on this determination.

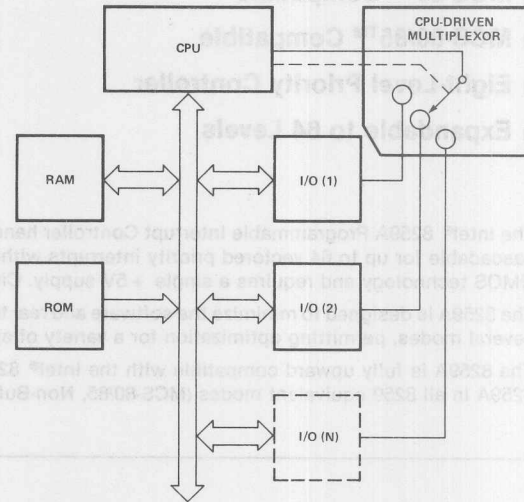
Each peripheral device or structure usually has a special program or "routine" that is associated with its specific functional or operational requirements; this is referred to as a "service routine". The PIC, after issuing an interrupt to the CPU, must somehow input information into the CPU that can "point" the Program Counter to the service routine associated with the requesting device. This "pointer" is an address in a vectoring table and will often be referred to, in this document, as vectoring data.

## 8259A BASIC FUNCTIONAL DESCRIPTION

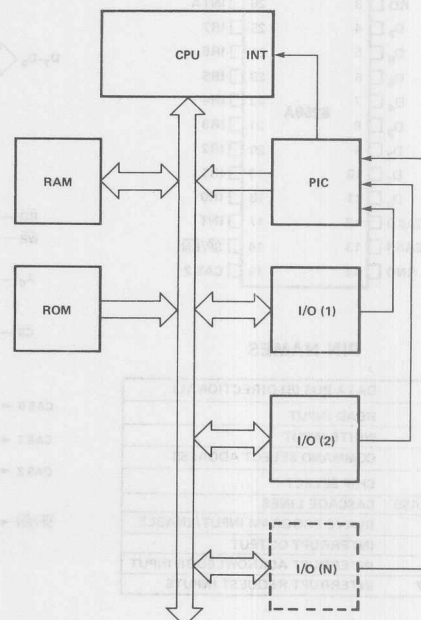
### GENERAL

The 8259A is a device specifically designed for use in real time, interrupt driven microcomputer systems. It manages eight levels or requests and has built-in features for expandability to other 8259A's (up to 64 levels). It is programmed by the system's software as an I/O peripheral. A selection of priority modes is available to the programmer so that the manner in which the requests are processed by the 8259A can be configured to

match his system requirements. The priority modes can be changed or reconfigured dynamically at any time during the main program. This means that the complete interrupt structure can be defined as required, based on the total system environment.



Polled Method



Interrupt Method

### INTERRUPT REQUEST REGISTER (IRR) AND IN-SERVICE REGISTER (ISR)

The Interrupts at the IR input lines are handled by two registers in cascade, the Interrupt Request Register (IRR) and the In-Service Register (ISR). The IRR is used to store all the interrupt levels which are requesting service, and the ISR is used to store all the interrupt levels which are being serviced.

### PRIORITY RESOLVER

This logic block determines the priorities of the bits set in the IRR. The highest priority is selected and strobed into the corresponding bit of the ISR during INTA pulse.

### INTERRUPT MASK REGISTER (IMR)

The IMR stores the bits which mask the interrupt lines to be masked. The IMR operates on the IRR. Masking of a higher priority input will not affect the interrupt request lines of lower priority.

### INT (INTERRUPT)

This output goes directly to the CPU interrupt input. The  $V_{OH}$  level on this line is designed to be fully compatible with the 8080A, 8085A and 8086 input levels.

### INTA (INTERRUPT ACKNOWLEDGE)

INTA pulses will cause the 8259A to release vectoring information onto the data bus. The format of this data depends on the system mode ( $\mu$ PM) of the 8259A.

### DATA BUS BUFFER

This 3-state, bidirectional 8-bit buffer is used to interface the 8259A to the system Data Bus. Control words and status information are transferred through the Data Bus Buffer.

### READ/WRITE CONTROL LOGIC

The function of this block is to accept OUTPUT commands from the CPU. It contains the Initialization Command Word (ICW) registers and Operation Command Word (OCW) registers which store the various control formats for device operation. This function block also allows the status of the 8259A to be transferred onto the Data Bus.

### $\overline{CS}$ (CHIP SELECT)

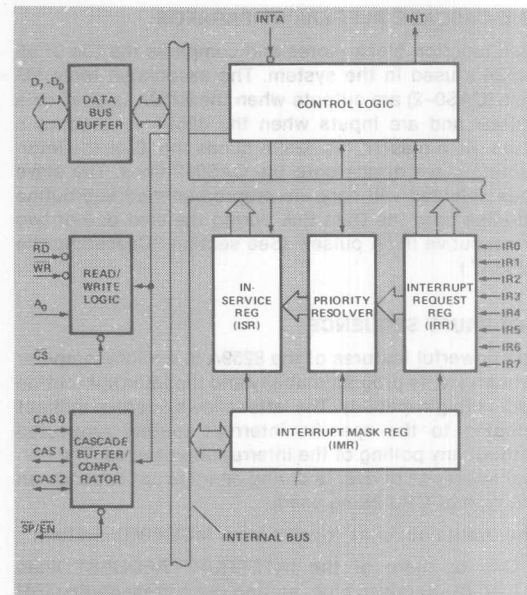
A LOW on this input enables the 8259A. No reading or writing of the chip will occur unless the device is selected.

### $\overline{WR}$ (WRITE)

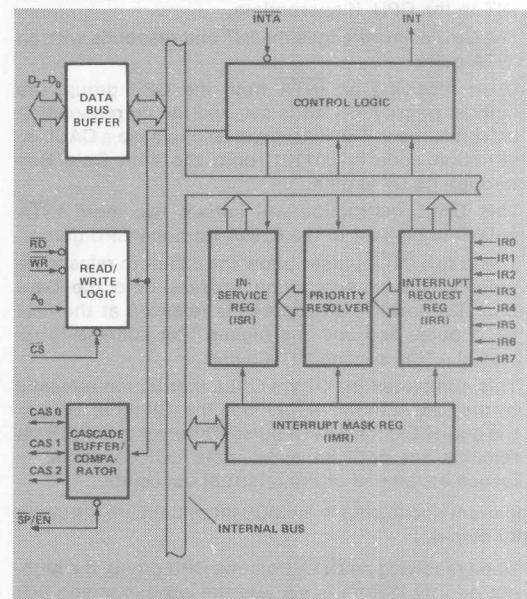
A LOW on this input enables the CPU to write control words (ICWs and OCWs) to the 8259A.

### $\overline{RD}$ (READ)

A LOW on this input enables the 8259A to send the status of the Interrupt Request Register (IRR), In Service Register (ISR), the Interrupt Mask Register (IMR), or the Interrupt level onto the Data Bus.



8259A Block Diagram



8259A Block Diagram

### A<sub>0</sub>

This input signal is used in conjunction with  $\overline{WR}$  and  $\overline{RD}$  signals to write commands into the various command registers, as well as reading the various status registers of the chip. This line can be tied directly to one of the address lines.

### THE CASCADE BUFFER/COMPARATOR

This function block stores and compares the IDs of all 8259A's used in the system. The associated three I/O pins (CAS0-2) are outputs when the 8259A is used as a master and are inputs when the 8259A is used as a slave. As a master, the 8259A sends the ID of the interrupting slave device onto the CAS0-2 lines. The slave thus selected will send its preprogrammed subroutine address onto the Data Bus during the next one or two consecutive  $\overline{\text{INTA}}$  pulses. (See section "Cascading the 8259A".)

### INTERRUPT SEQUENCE

The powerful features of the 8259A in a microcomputer system are its programmability and the interrupt routine addressing capability. The latter allows direct or indirect jumping to the specific interrupt routine requested without any polling of the interrupting devices. The normal sequence of events during an interrupt depends on the type of CPU being used.

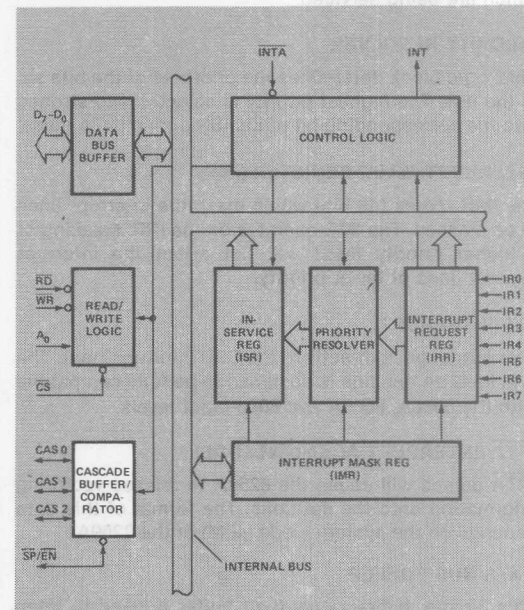
The events occur as follows in an MCS-80/85 system:

1. One or more of the INTERRUPT REQUEST lines (IR7-0) are raised high, setting the corresponding IRR bit(s).
2. The 8259A evaluates these requests, and sends an INT to the CPU, if appropriate.
3. The CPU acknowledges the INT and responds with an  $\overline{\text{INTA}}$  pulse.
4. Upon receiving an  $\overline{\text{INTA}}$  from the CPU group, the highest priority ISR bit is set, and the corresponding IRR bit is reset. The 8259A will also release a CALL instruction code (11001101) onto the 8-bit Data Bus through its D7-0 pins.
5. This CALL instruction will initiate two more  $\overline{\text{INTA}}$  pulses to be sent to the 8259A from the CPU group.
6. These two  $\overline{\text{INTA}}$  pulses allow the 8259A to release its preprogrammed subroutine address onto the Data Bus. The lower 8-bit address is released at the first  $\overline{\text{INTA}}$  pulse and the higher 8-bit address is released at the second  $\overline{\text{INTA}}$  pulse.
7. This completes the 3-byte CALL instruction released by the 8259A. In the AEOL mode the ISR bit is reset at the end of the third  $\overline{\text{INTA}}$  pulse. Otherwise, the ISR bit remains set until an appropriate EOI command is issued at the end of the interrupt sequence.

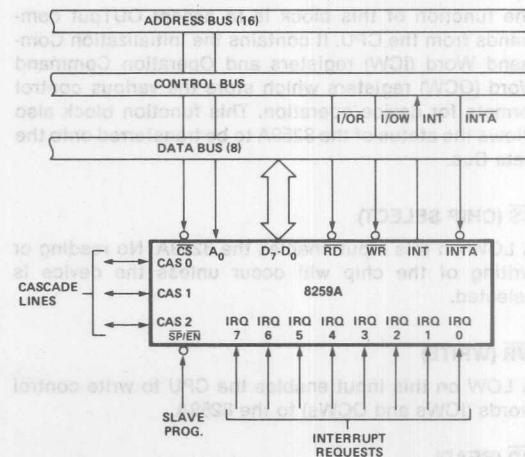
The events occurring in an MCS-86 system are the same until step 4.

4. Upon receiving an  $\overline{\text{INTA}}$  from the CPU group, the highest priority ISR bit is set and the corresponding IRR bit is reset. The 8259A does not drive the Data Bus during this cycle.
5. The MCS-86 CPU will initiate a second  $\overline{\text{INTA}}$  pulse. During this pulse, the 8259A releases an 8-bit pointer onto the Data Bus where it is read by the CPU.
6. This completes the interrupt cycle. In the AEOL mode the ISR bit is reset at the end of the second  $\overline{\text{INTA}}$  pulse. Otherwise, the ISR bit remains set until an appropriate EOI command is issued at the end of the interrupt subroutine.

If no interrupt request is present at step 4 of either sequence (i.e., the request was too short in duration) the 8259A will issue an interrupt level 7. Both the vectoring bytes and the CAS lines will look like an interrupt level 7 was requested.



8259A Block Diagram



8259A Interface to Standard System Bus

## INTERRUPT SEQUENCE OUTPUTS

### MCS-80/85 SYSTEM

This sequence is timed by three  $\overline{\text{INTA}}$  pulses. During the first  $\overline{\text{INTA}}$  pulse the CALL opcode is enabled onto the data bus.

Content of First Interrupt  
Vector Byte

	D7	D6	D5	D4	D3	D2	D1	D0
CALL CODE	1	1	0	0	1	1	0	1

During the second  $\overline{\text{INTA}}$  pulse the lower address of the appropriate service routine is enabled onto the data bus. When Interval = 4 bits A<sub>5</sub>-A<sub>7</sub> are programmed, while A<sub>0</sub>-A<sub>4</sub> are automatically inserted by the 8259A. When Interval = 8 only A<sub>6</sub> and A<sub>7</sub> are programmed, while A<sub>0</sub>-A<sub>5</sub> are automatically inserted.

Content of Second Interrupt  
Vector Byte

IR	Interval = 4							
	D7	D6	D5	D4	D3	D2	D1	D0
7	A7	A6	A5	1	1	1	0	0
6	A7	A6	A5	1	1	0	0	0
5	A7	A6	A5	1	0	1	0	0
4	A7	A6	A5	1	0	0	0	0
3	A7	A6	A5	0	1	1	0	0
2	A7	A6	A5	0	1	0	0	0
1	A7	A6	A5	0	0	1	0	0
0	A7	A6	A5	0	0	0	0	0

IR	Interval = 8							
	D7	D6	D5	D4	D3	D2	D1	D0
7	A7	A6	1	1	1	0	0	0
6	A7	A6	1	1	0	0	0	0
5	A7	A6	1	0	1	0	0	0
4	A7	A6	1	0	0	0	0	0
3	A7	A6	0	1	1	0	0	0
2	A7	A6	0	1	0	0	0	0
1	A7	A6	0	0	1	0	0	0
0	A7	A6	0	0	0	0	0	0

During the third  $\overline{\text{INTA}}$  pulse the higher address of the appropriate service routine, which was programmed as byte 2 of the initialization sequence (A<sub>8</sub>-A<sub>15</sub>), is enabled onto the bus.

Content of Third Interrupt  
Vector Byte

D7	D6	D5	D4	D3	D2	D1	D0
A15	A14	A13	A12	A11	A10	A9	A8

### MCS-86 SYSTEM

MCS-86 mode is similar to MCS-80 mode except that only two Interrupt Acknowledge cycles are issued by the processor and no CALL opcode is sent to the processor. The first interrupt acknowledge cycle is similar to that of MCS-80/85 systems in that the 8259A uses it to internally freeze the state of the interrupts for priority resolution and as a master it issues the interrupt code on the cascade lines at the end of the  $\overline{\text{INTA}}$  pulse. On this first cycle it does not issue any data to the processor and leaves its data bus buffers disabled. On the second interrupt acknowledge cycle in MCS-86 mode the master (or slave if so programmed) will send a byte of data to the processor with the acknowledged interrupt code composed as follows (note the state of the ADI mode control is ignored and A<sub>5</sub>-A<sub>11</sub> are unused in MCS-86 mode):

Content of Interrupt Vector Byte  
for MCS-86 System Mode

	D7	D6	D5	D4	D3	D2	D1	D0
IR7	A15	A14	A13	A12	A11	1	1	1
IR6	A15	A14	A13	A12	A11	1	1	0
IR5	A15	A14	A13	A12	A11	1	0	1
IR4	A15	A14	A13	A12	A11	1	0	0
IR3	A15	A14	A13	A12	A11	0	1	1
IR2	A15	A14	A13	A12	A11	0	1	0
IR1	A15	A14	A13	A12	A11	0	0	1
IR0	A15	A14	A13	A12	A11	0	0	0

ated by the CPU:

1. **Initialization Command Words (ICWs):** Before normal operation can begin, each 8259A in the system must be brought to a starting point — by a sequence of 2 to 4 bytes timed by  $\overline{WR}$  pulses. This sequence is described in Figure 1.

2. **Operation Command Words (OCWs):** These are the command words which command the 8259A to operate in various interrupt modes. These modes are:

- Fully nested mode
- Rotating priority mode
- Special mask mode
- Polled mode

The OCWs can be written into the 8259A anytime after initialization.

## GENERAL

Whenever a command is issued with  $A_0 = 0$  and  $D_4 = 1$ , this is interpreted as Initialization Command Word 1 (ICW1). ICW1 starts the initialization sequence during which the following automatically occur.

- The Interrupt Mask Register is cleared.
- IR 7 input is assigned priority 7.
- The slave mode address is set to 7.
- Special Mask Mode is cleared and Status Read is set to IRR.
- If  $IC_4 = 0$ , then all functions selected in ICW4 are set to zero. (Non-Buffered mode\*, no Auto-EOI, MCS-80/85 system, non SFNM).

\*Note: Master/Slave in ICW4 is only used in the buffered mode.

A = address							
00	01	02	03	04	05	06	07
0	0	1	1	1	0A	0A	7A
0	0	0	1	1	0A	0A	7A
0	0	1	0	1	0A	0A	7A
0	0	0	0	1	0A	0A	7A
0	0	1	1	0	0A	0A	7A
0	0	0	1	0	0A	0A	7A
0	0	0	0	0	0A	0A	7A

$A_0$	$D_4$	$D_3$	$\overline{RD}$	$\overline{WR}$	$\overline{CS}$	INPUT OPERATION (READ)
0			0	1	0	IRR, ISR or Interrupting Level → DATA BUS (Note 1)
1			0	1	0	IMR → DATA BUS
						OUTPUT OPERATION (WRITE)
0	0	0	1	0	0	DATA BUS → OCW2
0	0	1	1	0	0	DATA BUS → OCW3
0	1	X	1	0	0	DATA BUS → ICW1
1	X	X	1	0	0	DATA BUS → OCW1, ICW2, ICW3, ICW4 (Note 2)
						DISABLE FUNCTION
X	X	X	1	1	0	DATA BUS — 3-STATE (NO OPERATION)
X	X	X	X	X	1	DATA BUS — 3-STATE (NO OPERATION)

Notes: 1. Selection of IRR, ISR or Interrupting Level is based on the content of OCW3 written before the READ operation.

2. On-chip sequencer logic queues these commands into proper sequence.

## 8259A Basic Operation

### INITIALIZATION COMMAND WORDS 1 AND 2 (ICW1, ICW2)

**A<sub>5</sub>-A<sub>15</sub>:** Page starting address of service routines. In an MCS 80/85 system, the 8 request levels will generate CALLs to 8 locations equally spaced in memory. These can be programmed to be spaced at intervals of 4 or 8 memory locations, thus the 8 routines will occupy a page of 32 or 64 bytes, respectively.

The address format is 2 bytes long (A<sub>0</sub>-A<sub>15</sub>). When the routine interval is 4, A<sub>0</sub>-A<sub>4</sub> are automatically inserted by the 8259A, while A<sub>5</sub>-A<sub>15</sub> are programmed externally. When the routine interval is 8, A<sub>0</sub>-A<sub>5</sub> are automatically inserted by the 8259A, while A<sub>6</sub>-A<sub>15</sub> are programmed externally.

The 8-byte interval will maintain compatibility with current software, while the 4-byte interval is best for a compact jump table.

In an MCS-86 system A<sub>15</sub>-A<sub>11</sub> are inserted in the five most significant bits of the vectoring byte and the 8259A sets the three least significant bits according to the interrupt level. A<sub>10</sub>-A<sub>5</sub> are ignored and ADI (Address interval) has no effect.

**LTIM:** If LTIM = 1, then the 8259A will operate in the level interrupt mode. Edge detect logic on the interrupt inputs will be disabled.

**ADI:** CALL address interval. ADI = 1 then interval = 4; ADI = 0 then interval = 8.

**SNGL:** Single. Means that this is the only 8259A in the system. If SNGL = 1 no ICW3 will be issued.

**IC4:** If this bit is set — ICW4 has to be read. If ICW4 is not needed, set IC4 = 0.

### INITIALIZATION COMMAND WORD 3 (ICW3)

This word is read only when there is more than one 8259A in the system and cascading is used, in which case SNGL = 0. It will load the 8-bit slave register. The functions of this register are:

- In the master mode (either when  $\overline{SP} = 1$ , or in buffered mode when M/S = 1 in ICW4) a "1" is set for each slave in the system. The master then will release byte 1 of the call sequence (for MCS-80/85 system) and will enable the corresponding slave to release bytes 2 and 3 (for MCS-86 only byte 2) through the cascade lines.
- In the slave mode (either when  $\overline{SP} = 0$ , or if BUF = 1 and M/S = 0 in ICW4) bits 2-0 identify the slave. The slave compares its cascade input with these bits and, if they are equal, bytes 2 and 3 of the call sequence (or just byte 2 for MCS-86) are released by it on the Data Bus.

### INITIALIZATION COMMAND WORD 4 (ICW4)

**SFNM:** If SFNM = 1 the special fully nested mode is programmed.

**BUF:** If BUF = 1 the buffered mode is programmed. In buffered mode  $\overline{SP}/\overline{EN}$  becomes an enable output and the master/slave determination is by M/S.

**M/S:** If buffered mode is selected: M/S = 1 means the 8259A is programmed to be a master, M/S = 0 means the 8259A is programmed to be a slave. If BUF = 0, M/S has no function.

**AEOI:** If AEOI = 1 the automatic end of interrupt mode is programmed.

**$\mu$ PM:** Microprocessor mode:  $\mu$ PM = 0 sets the 8259A for MCS-80/85 system operation,  $\mu$ PM = 1 sets the 8259A for MCS-86 system operation.

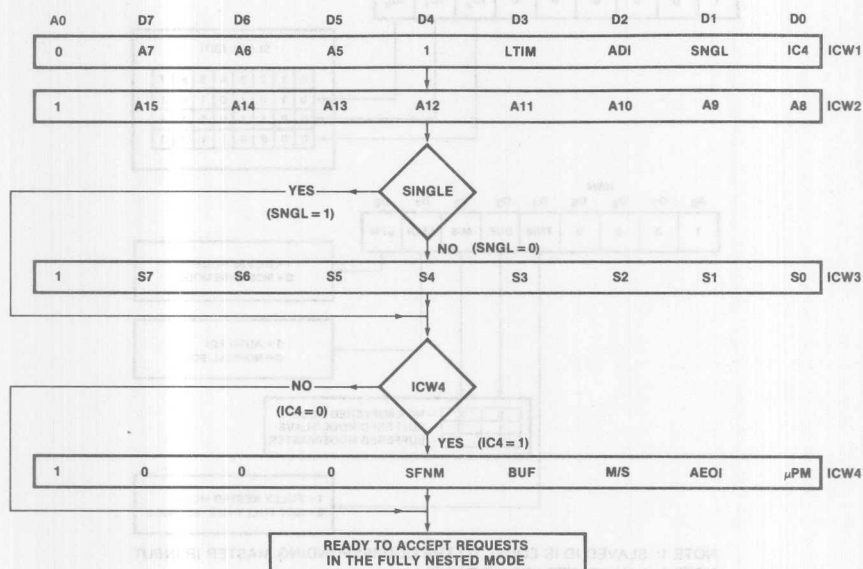
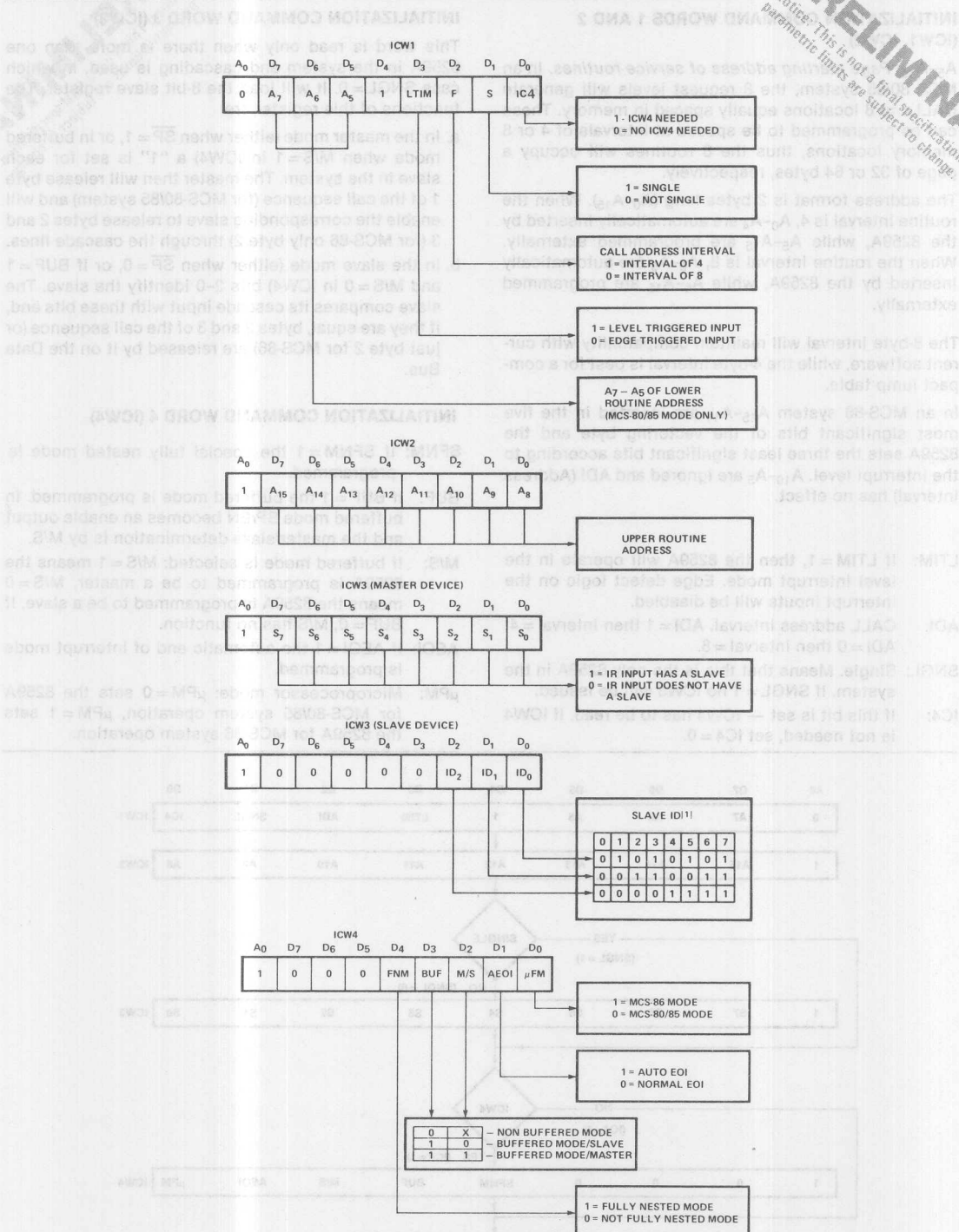


Figure 1. Initialization Sequence

**PRELIMINARY**  
 Notice: This is not a final specification. Some parametric limits are subject to change.



NOTE 1: SLAVED ID IS EQUAL TO THE CORRESPONDING MASTER IR INPUT.  
 NOTE 2: X INDICATED "DON'T CARE".

## OPERATION COMMAND WORDS (OCWs)

After the Initialization Command Words (ICWs) are programmed into the 8259A, the chip is ready to accept interrupt requests at its input lines. However, during the 8259A operation, a selection of algorithms can command the 8259A to operate in various modes through the Operation Command Words (OCWs).

### OPERATION CONTROL WORDS (OCWs)

A0	OCW1							
	D7	D6	D5	D4	D3	D2	D1	D0
1	M7	M6	M5	M4	M3	M2	M1	M0

A0	OCW2							
	R	SEOI	EOI	0	0	L2	L1	L0
0								

A0	OCW3							
	0	SSMM	SMM	0	1	P	SRIS	RIS
0								

### OPERATION CONTROL WORD 1 (OCW1)

OCW1 sets and clears the mask bits in the Interrupt Mask Register (IMR). M<sub>7</sub>–M<sub>0</sub> represent the eight mask bits. M = 1 indicates the channel is masked (inhibited). M = 0 indicates the channel is enabled.

### OPERATION CONTROL WORD 2 (OCW2)

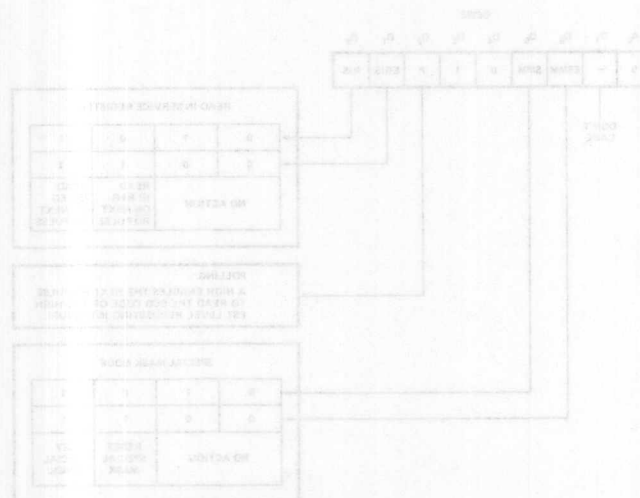
R, SEOI, EOI — These three bits control the Rotate and End of Interrupt modes and combinations of the two. A chart of these combinations can be found on the Operation Command Word Format.

L<sub>2</sub>, L<sub>1</sub>, L<sub>0</sub> — These bits determine the interrupt level acted upon when the SEOI bit is active.

### OPERATION CONTROL WORD 3 (OCW3)

ESMM — Enable Special Mask Mode. When this bit is set to 1 it enables the SMM bit to set or reset the Special Mask Mode. When ESMM = 0 the SMM bit becomes a "don't care".

SMM — Special Mask Mode. If ESMM = 1 and SMM = 1 the 8259A will enter Special Mask Mode. If ESMM = 1 and SMM = 0 the 8259A will revert to normal mask mode. When ESMM = 0, SMM has no effect.



1	M7	M6	M5	M4	M3	M2	M1	M0
---	----	----	----	----	----	----	----	----

INTERRUPT MASK  
1 = MASK SET  
0 = MASK RESET

OPERATION CONTROL WORD 2 (OCW2)  
R, SEOI, EOI — These bits control the Rotate and End of Interrupt modes and combinations of the two. A combination of the two can be found in the Operation Command Word 3 (OCW3).

A <sub>0</sub>	D <sub>7</sub>	D <sub>6</sub>	D <sub>5</sub>	D <sub>4</sub>	D <sub>3</sub>	D <sub>2</sub>	D <sub>1</sub>	D <sub>0</sub>
0	R	SEOI	EOI	0	0	L <sub>2</sub>	L <sub>1</sub>	L <sub>0</sub>

BCD LEVEL TO BE RESET OR PUT INTO LOWEST PRIORITY							
0	1	2	3	4	5	6	7
0	1	0	1	0	1	0	1
0	0	1	1	0	0	1	1
0	0	0	0	1	1	1	1

0	0	1	Non-specific EOI
0	1	1	Specific EOI. L0-L2 code of IS FF to be reset
1	0	1	Rotate at EOI Automatically (Mode A)
1	1	1	Rotate at EOI (mode B). L0-L2 code of line
1	0	0	Set Rotate A FF
0	0	0	Clear Rotate A FF
1	1	0	Rotate priority (mode B) independently of EOI
0	1	0	No Operation

A <sub>0</sub>	D <sub>7</sub>	D <sub>6</sub>	D <sub>5</sub>	D <sub>4</sub>	D <sub>3</sub>	D <sub>2</sub>	D <sub>1</sub>	D <sub>0</sub>
0	—	ESMM	SMM	0	1	P	ERIS	RIS

DON'T CARE

READ IN SERVICE REGISTER			
0	1	0	1
0	0	1	1
NO ACTION	READ IR REG ON NEXT RD PULSE	READ IS REG ON NEXT RD PULSE	

POLLING  
A HIGH ENABLES THE NEXT RD PULSE TO READ THE BCD CODE OF THE HIGHEST LEVEL REQUESTING INTERRUPT.

SPECIAL MASK MODE			
0	1	0	1
0	0	1	1
NO ACTION	RESET SPECIAL MASK	SET SPECIAL MASK	

## INTERRUPT MASKS

Each Interrupt Request input can be masked individually by the Interrupt Mask Register (IMR) programmed through OCW1. Each bit in the IMR masks one interrupt channel if it is set (1). Bit 0 masks IR0, Bit 1 masks IR1 and so forth. Masking an IR channel does not affect the other channels operation.

## SPECIAL MASK MODE

Some applications may require an interrupt service routine to dynamically alter the system priority structure during its execution under software control. For example, the routine may wish to inhibit lower priority requests for a portion of its execution but enable some of them for another portion.

The difficulty here is that if an Interrupt Request is acknowledged and an End of Interrupt command did not reset its IS bit (i.e., while executing a service routine), the 8259A would have inhibited all lower priority requests with no easy way for the routine to enable them.

That is where the Special Mask Mode comes in. In the special Mask Mode, when a mask bit is set in OCW1, it inhibits further interrupts at that level and enables interrupts from all other levels (lower as well as higher) that are not masked.

Thus, any interrupts may be selectively enabled by loading the mask register.

The special Mask Mode is set by OCW3 where: SSMM=1, SMM=1, and cleared where SSMM=1, SMM=0.

## BUFFERED MODE

When the 8259A is used in a large system where bus driving buffers are required on the data bus and the cascading mode is used, there exists the problem of enabling buffers.

The buffered mode will structure the 8259A to send an enable signal on  $\overline{SP/EN}$  to enable the buffers. In this mode, whenever the 8259A's data bus outputs are enabled, the  $\overline{SP/EN}$  output becomes active.

This modification forces the use of software programming to determine whether the 8259A is a master or a slave. Bit 3 in ICW4 programs the buffered mode, and bit 2 in ICW4 determines whether it is a master or a slave.

## NESTED MODE

This mode is entered after initialization unless another mode is programmed. The interrupt requests are ordered in priority from 0 through 7 (0 highest). When an interrupt is acknowledged the highest priority request is determined and its vector placed on the bus. Additionally, a bit of the Interrupt Service register (ISR-7) is set. This bit remains set until the microprocessor issues an End of Interrupt (EOI) command immediately before returning from the service routine, or if AEOI (Automatic End of Interrupt) bit is set, until the trailing edge of the last INTA. While the IS bit is set, all further interrupts of the same or lower priority are inhibited, while higher levels will generate an interrupt (which will be acknowledged only if the microprocessor internal Interrupt enable flip-flop has been re-enabled through software).

After the initialization sequence, IR0 has the highest priority and IR7 the lowest. Priorities can be changed, as will be explained, in the rotating priority mode.

## THE SPECIAL FULLY NESTED MODE

This mode will be used in the case of a big system where cascading is used, and the priority has to be conserved within each slave. In this case the fully nested mode will be programmed to the master (using ICW4). This mode is similar to the normal nested mode with the following exceptions:

- When an interrupt request from a certain slave is in service this slave is not locked out from the master's priority logic and further interrupt requests from higher priority IR's within the slave will be recognized by the master and will initiate interrupts to the processor. (In the normal nested mode a slave is masked out when its request is in service and no higher requests from the same slave can be serviced.)
- When exiting the Interrupt Service routine the software has to check whether the interrupt serviced was the only one from that slave. This is done by sending a non-specific End of Interrupt (EOI) command to the slave and then reading its In-Service register and checking for zero. If it is empty, a non-specific EOI can be sent to the master too. If not, no EOI should be sent.

## POLL

In this mode the microprocessor internal Interrupt Enable flip-flop is reset, disabling its interrupt input. Service to devices is achieved by programmer initiative using a Poll command.

The Poll command is issued by setting  $P = "1"$  in OCW3. The 8259A treats the next  $\overline{RD}$  pulse to the 8259A (i.e.,  $\overline{RD} = 0$ ,  $\overline{CS} = 0$ ) as an interrupt acknowledge, sets the appropriate IS bit if there is a request, and reads the priority level. Interrupt is frozen from  $\overline{WR}$  to  $\overline{RD}$ .

The word enabled onto the data bus during  $\overline{RD}$  is:

D7	D6	D5	D4	D3	D2	D1	D0
1	—	—	—	—	W2	W1	W0

W0-W2: Binary code of the highest priority level requesting service.

1: Equal to a "1" if there is an interrupt.

This mode is useful if there is a routine command common to several levels so that the  $\overline{INTA}$  sequence is not needed (saves ROM space). Another application is to use the poll mode to expand the number of priority levels to more than 64.

## END OF INTERRUPT (EOI)

The In Service (IS) bit can be reset either automatically following the trailing edge of the last in sequence  $\overline{INTA}$  pulse (when AEIOI bit in ICW1 is set) or by a command word that must be issued to the 8259A before returning from a service routine (EOI command). An EOI command must be issued twice, once for the master and once for the corresponding slave if slaves are in use.

There are two forms of EOI command: Specific and Non-Specific. When the 8259A is operated in modes which preserve the fully nested structure, it can determine which IS bit to reset on EOI. When a Non-Specific EOI command is issued the 8259A will automatically reset the highest IS bit of those that are set, since in the nested mode the highest IS level was necessarily the last level acknowledged and serviced.

However, when a mode is used which may disturb the fully nested structure, the 8259A may no longer be able to determine the last level acknowledged. In this case a Specific End of Interrupt (SEOI) must be issued which includes as part of the command the IS level to be reset. EOI is issued whenever  $EOI = 1$ , in OCW2, where L0-L2 is the binary level of the IS bit to be reset. Note that although the Rotate command can be issued together with an EOI where  $EOI = 1$ , it is not necessarily tied to it.

It should be noted that an IS bit that is masked by an IMR bit will not be cleared by a non-specific EOI if the 8259A is in the Special Mask Mode.

## AUTOMATIC END OF INTERRUPT (AEIOI) MODE

If AEIOI = 1 in ICW4, then the 8259A will operate in AEIOI mode continuously until reprogrammed by ICW4. In this mode the 8259A will automatically perform a non-specific EOI operation at the trailing edge of the last interrupt acknowledge pulse (third pulse in MCS-80/85,

second in MCS-86). Note that from a system standpoint, this mode should be used only when a nested multilevel interrupt structure is not required within a single 8259A.

To achieve automatic rotation (Rotate Mode A) within AEIOI, there is a special rotate flip-flop. It is set by OCW2 with  $R = 1$ , SEOI = 0, EOI = 0, and cleared with  $R = 0$ , SEOI = 0, EOI = 0.

## ROTATING PRIORITY MODE A (AUTOMATIC ROTATION) FOR EQUAL PRIORITY DEVICES

In some applications there are a number of interrupting devices of equal priority. In this mode a device, after being serviced, receives the lowest priority, so a device requesting an interrupt will have to wait, in the worst case until each of 7 other devices are serviced at most once. For example, if the priority and "in service" status is:

**Before Rotate** (IR4 the highest priority requiring service)

IS7	IS6	IS5	IS4	IS3	IS2	IS1	IS0
0	1	0	1	0	0	0	0
Lowest Priority				Highest Priority			
7	6	5	4	3	2	1	0

**After Rotate** (IR4 was serviced, all other priorities rotated correspondingly)

IS7	IS6	IS5	IS4	IS3	IS2	IS1	IS0
0	1	0	0	0	0	0	0
Highest Priority				Lowest Priority			
2	1	0	7	6	5	4	3

The Rotate command mode A is issued in OCW2 where:  $R = 1$ ,  $EOI = 1$ ,  $SEOI = 0$ . Internal status is updated by an End of Interrupt (EOI or AEIOI) command. If  $R = 1$ ,  $EOI = 0$ ,  $SEOI = 0$ , a "Rotate-A" flip-flop is set. This is useful in AEIOI, and described under Automatic End of Interrupt.

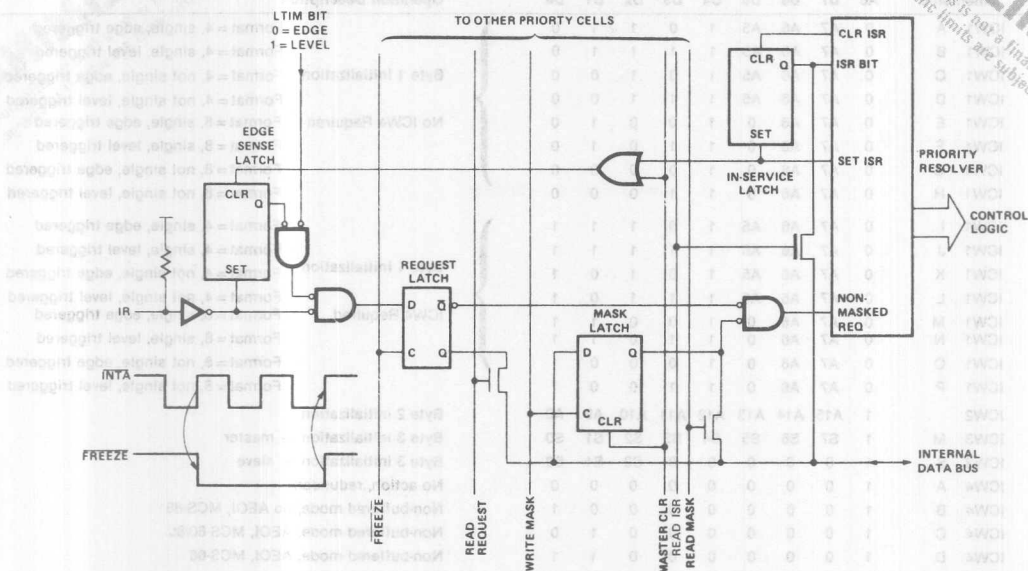
## ROTATING PRIORITY MODE B (ROTATION BY SOFTWARE)

The programmer can change priorities by programming the bottom priority and thus fixing all other priorities; i.e., if IR5 is programmed as the bottom priority device, then IR6 will have the highest one.

The Rotate command is issued in OCW2 where:  $R = 1$ ,  $SEOI = 1$ ; L0-L2 is the binary priority level code of the bottom priority device.

Observe that in this mode internal status is updated by software control during OCW2. However, it is independent of the End of Interrupt (EOI) command (also executed by OCW2). Priority changes can be executed during an EOI command or independently.

**PRELIMINARY**  
 Notice: This is not a final specification. Some  
 parametric limits are subject to change.



## NOTES

1. MASTER CLEAR ACTIVE ONLY DURING ICW1
2. FREEZE/IS ACTIVE DURING INTA AND POLL SEQUENCES ONLY
3. TRUTH TABLE FOR D-LATCH

C	D	Q	OPERATION
1	Di	Di	FOLLOW
0	X	Qn-1	HOLD

## Priority Cell — Simplified Logic Diagram

## LEVEL TRIGGERED MODE

This mode is programmed using bit 3 in ICW1.

If LTIM = '1', an interrupt request will be recognized by a 'high' level on IR input, and there is no need for an edge detection. The interrupt request must be removed before the EOI command is issued or the CPU interrupt is enabled to prevent a second interrupt from occurring.

The above figure shows a conceptual circuit to give the reader an understanding of the level sensitive and edge sensitive input circuitry of the 8259A. Be sure to note that the request latch is a transparent D type latch.

## READING THE 8259A STATUS

The input status of several internal registers can be read to update the user information on the system. The following registers can be read by issuing a suitable OCW3 and reading with  $\overline{RD}$ .

**Interrupt Mask Register:** 8-bit register whose content specifies the interrupt request lines being masked. acknowledged. The highest request level is reset from the IRR when an interrupt is acknowledged. (Not affected by IMR.)

**In-Service Register (ISR):** 8-bit register which contains the priority levels that are being serviced. The ISR is updated when an End of Interrupt command is issued.

**Interrupt Mask Register:** 8-bit register which contains the interrupt request lines which are masked.

The IRR can be read when, prior to the  $\overline{RD}$  pulse, a  $\overline{WR}$  pulse is issued with OCW3 (ERIS = 1, RIS = 0.)

The ISR can be read in a similar mode when ERIS = 1, RIS = 1 in the OCW3.

There is no need to write an OCW3 before every status read operation, as long as the status read corresponds with the previous one; i.e., the 8259A "remembers" whether the IRR or ISR has been previously selected by the OCW3. This is not true when poll is used.

After initialization the 8259A is set to IRR.

For reading the IMR, no OCW3 is needed. The output data bus will contain the IMR whenever  $\overline{RD}$  is active and A0 = 1.

Polling overrides status read when P = 1, ERIS = 1 in OCW3.

3	ICW1	C	0	A7	A6	A5	1	1	1	0	0		Byte 1 initialization	Format = 4, not single, edge triggered
4	ICW1	D	0	A7	A6	A5	1	1	1	0	0		No ICW4 Required	Format = 4, not single, level triggered
5	ICW1	E	0	A7	A6	0	1	0	0	1	0			Format = 8, single, edge triggered
6	ICW1	F	0	A7	A6	0	1	1	0	1	0			Format = 8, single, level triggered
7	ICW1	G	0	A7	A6	0	1	0	0	0	0			Format = 8, not single, edge triggered
8	ICW1	H	0	A7	A6	0	1	1	0	0	0			Format = 8, not single, level triggered
9	ICW1	I	0	A7	A6	A5	1	0	1	1	1		Byte 1 Initialization	Format = 4, single, edge triggered
10	ICW1	J	0	A7	A6	A5	1	1	1	1	1			Format = 4, single, level triggered
11	ICW1	K	0	A7	A6	A5	1	0	1	0	1			Format = 4, not single, edge triggered
12	ICW1	L	0	A7	A6	A5	1	1	1	0	1			Format = 4, not single, level triggered
13	ICW1	M	0	A7	A6	0	1	0	0	1	1		ICW4 Required	Format = 8, single, edge triggered
14	ICW1	N	0	A7	A6	0	1	1	0	1	1			Format = 8, single, level triggered
15	ICW1	O	0	A7	A6	0	1	0	0	0	1			Format = 8, not single, edge triggered
16	ICW1	P	0	A7	A6	0	1	1	0	0	1			Format = 8, not single, level triggered
17	ICW2		1	A15	A14	A13	A12	A11	A10	A9	A8		Byte 2 initialization	
18	ICW3	M	1	S7	S6	S5	S4	S3	S2	S1	S0		Byte 3 initialization — master	
19	ICW3	S	1	0	0	0	0	0	S2	S1	S0		Byte 3 initialization — slave	
20	ICW4	A	1	0	0	0	0	0	0	0	0		No action, redundant	
21	ICW4	B	1	0	0	0	0	0	0	0	1		Non-buffered mode, no AEOI, MCS-86	
22	ICW4	C	1	0	0	0	0	0	0	1	0		Non-buffered mode, AEOI, MCS-80/85	
23	ICW4	D	1	0	0	0	0	0	0	1	1		Non-buffered mode, AEOI, MCS-86	
24	ICW4	E	1	0	0	0	0	0	1	0	0		No action, redundant	
25	ICW4	F	1	0	0	0	0	0	1	0	1		Non-buffered mode, no AEOI, MCS-86	
26	ICW4	G	1	0	0	0	0	0	1	1	0		Non-buffered mode, AEOI, MCS-80/85	
27	ICW4	H	1	0	0	0	0	0	1	1	1		Non-buffered mode, AEOI, MCS-86	
28	ICW4	I	1	0	0	0	0	1	0	0	0		Buffered mode, slave, no AEOI, MCS-80/85	
29	ICW4	J	1	0	0	0	0	1	0	0	1		Buffered mode, slave, no AEOI, MCS-86	
30	ICW4	K	1	0	0	0	0	1	0	1	0		Buffered mode, slave, AEOI, MCS-80/85	
31	ICW4	L	1	0	0	0	0	1	0	1	1		Buffered mode, slave, AEOI, MCS-86	
32	ICW4	M	1	0	0	0	0	1	1	0	0		Buffered mode, master, no AEOI, MCS-80/85	
33	ICW4	N	1	0	0	0	0	1	1	0	1		Buffered mode, master, no AEOI, MCS-86	
34	ICW4	O	1	0	0	0	0	1	1	1	0		Buffered mode, master, AEOI, MCS-80/85	
35	ICW4	P	1	0	0	0	0	1	1	1	1		Buffered mode, master, AEOI, MCS-86	
36	ICW4	NA	1	0	0	0	1	0	0	0	0		Fully nested mode, MCS-80, non-buffered, no AEOI	
37	ICW4	NB	1	0	0	0	1	0	0	0	1		ICW4 NB through ICW4 ND are identical to ICW4 B through ICW4 D with the addition of Fully Nested Mode	
38	ICW4	NC	1	0	0	0	1	0	0	1	0			
39	ICW4	ND	1	0	0	0	1	0	0	1	1			
40	ICW4	NE	1	0	0	0	1	0	1	0	0			Fully Nested Mode, MCS-80/85, non-buffered, no AEOI
41	ICW4	NF	1	0	0	0	1	0	1	0	1		ICW4 NF through ICW4 NP are identical to ICW4 F through ICW4 P with the addition of Fully Nested Mode	
42	ICW4	NG	1	0	0	0	1	0	1	1	0			
43	ICW4	NH	1	0	0	0	1	0	1	1	1			
44	ICW4	NI	1	0	0	0	1	1	0	0	0			
45	ICW4	NJ	1	0	0	0	1	1	0	0	1			
46	ICW4	NK	1	0	0	0	1	1	0	1	0			
47	ICW4	NL	1	0	0	0	1	1	0	1	1			
48	ICW4	NM	1	0	0	0	1	1	1	0	0			
49	ICW4	NN	1	0	0	0	1	1	1	0	1			
50	ICW4	NO	1	0	0	0	1	1	1	1	0			
51	ICW4	NP	1	0	0	0	1	1	1	1	1			
52	OCW1		1	M7	M6	M5	M4	M3	M2	M1	M0		Load mask register, read mask register	
53	OCW2	E	0	0	0	1	0	0	0	0	0		Non-specific EOI	
54	OCW2	SE	0	0	1	1	0	0	L2	L1	L0		Specific EOI. L0-L2 code of IS FF to be reset	
55	OCW2	RE	0	1	0	1	0	0	0	0	0		Rotate at EOI Automatically (Mode A)	
56	OCW2	RSE	0	1	1	1	0	0	L2	L1	L0		Rotate at EOI (mode B). L0-L2 code of line	
57	OCW2	R	0	1	0	0	0	0	0	0	0		Set Rotate A FF	
58	OCW2	CR	0	0	0	0	0	0	0	0	0		Clear Rotate A FF	
59	OCW2	RS	0	1	1	0	0	0	L2	L1	L0		Rotate priority (mode B) independently of EOI	
60	OCW3	P	0	0	0	0	0	1	1	0	0		Poll mode	
61	OCW3	RIS	0	0	0	0	0	1	0	1	1		Read IS register	

**Note:** 1. In the master mode  $\overline{SP}$  pin = 1, in slave mode  $\overline{SP}$  = 0

### Cascading

The 8259A can be easily interfaced to one master with up to eight slave processors, each with its own priority levels.

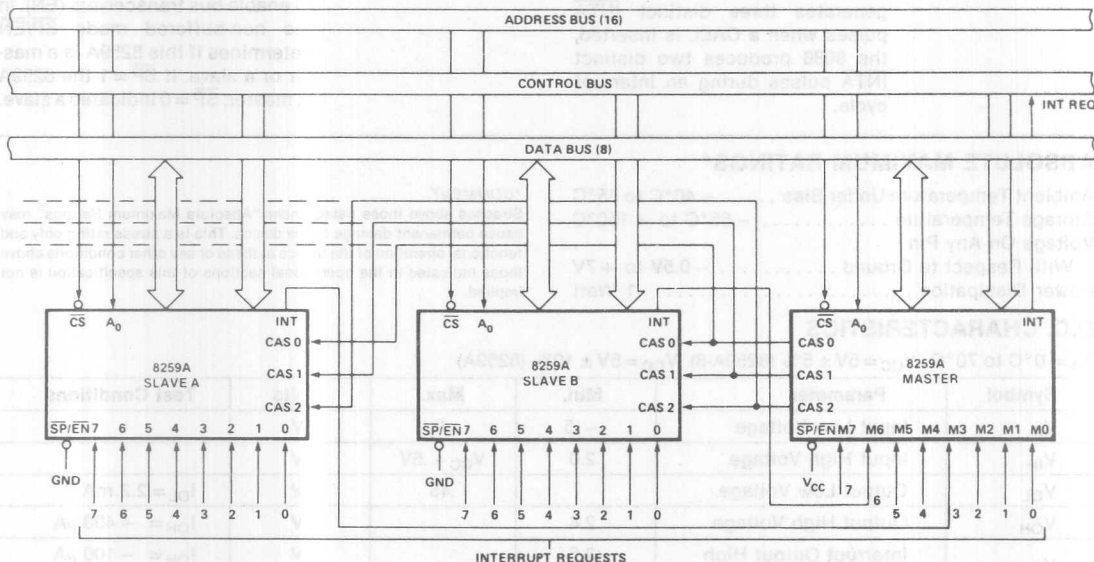
A typical MCS-80/85 system is master controls, through the 3 one of the slaves will release address.

As shown in Figure 2, the slave is connected to the master interrupt. When a slave request line is activated and the master interrupt is edge-triggered, the master will enable the slave to release the device routine address. (Byte 2 only for MCS-8).

not be connected to a slave 8:  
have slaves attached.

The cascade bus lines are normally initialized with the slave address code from the INTA pulse to the trailing edge of the INTA pulse. It is obvious that each 8259A in the system must be initialized in a separate initialization sequence. The initialization sequence must be programmed to work in a different manner for each slave. It must be issued twice: once for the master and once for the corresponding slave. An address code must be issued to activate the Chip Select (CS<sub>0</sub>) of the slave.

The cascade lines of the Master can be used as any interrupt input, even if no slave is connected to the input.



### Figure 2. Cascading the 8259A

## PIN FUNCTIONS

Name	I/O	Pin #	Function	CS	I	1	Chip Select: RD and WR are enabled by Chip Select, whereas Interrupt Acknowledge, is independent of Chip Select.
V <sub>CC</sub>		28	+5V supply.				
GND		14	Ground.				
D <sub>0-7</sub>	I/O	11-14	Bidirectional data bus, used for: a) programming the mode of the 8259A (programming is done by software); b) the microprocessor can read the status of the 8259A; c) the 8259A will send vectoring data to the microprocessor when an interrupt is acknowledged.	A0	I	27	Usually the least significant bit of the microprocessor address output. When A0=1 the Interrupt Mask Register can be loaded or read. When A0=0 the 8259A mode can be programmed or its status can be read. CS is active LOW.
IR <sub>0-7</sub>	I	18-25	Interrupt Requests: These are asynchronous inputs. A positive-going edge will generate an interrupt request. Thus a request can be generated by raising the line and holding it high until acknowledged, or by a negative pulse. In level triggered mode, no edge is required. These lines are active HIGH.	INT	O	17	Goes directly to the microprocessor interrupt input. This output will have high V <sub>OH</sub> to match the 8080 3.3V V <sub>IH</sub> . INT is active HIGH.
RD	I	3	Read (generally from 8228 in MCS-80 system or from 8086 in MCS-86 system).	C0-C2	I/O	12 13 15	Three cascade lines, outputs in master mode and inputs in slave mode. The master issues the binary code of the acknowledged interrupt level on these lines. Each slave compares this code with its own.
WR	I	2	Write (generally from 8228 in MCS-80 system or from 8086 in MCS-86 system).	SP/EN	I/O	16	SP/EN is a dual function pin. In the buffered mode SP/EN is used to enable bus transceivers (EN). In the non-buffered mode SP/EN determines if this 8259A is a master or a slave. If SP=1 the 8259A is master; SP=0 indicates a slave.
INTA	I	26	Interrupt Acknowledge (generally from 8228 in MCS-80 system, 8086 in MCS-86 system). The 8288 generates three distinct INTA pulses when a CALL is inserted, the 8086 produces two distinct INTA pulses during an interrupt cycle.				

## ABSOLUTE MAXIMUM RATINGS\*

Ambient Temperature Under Bias . . . . . -40°C to 85°C  
 Storage Temperature . . . . . -65°C to +150°C  
 Voltage On Any Pin

With Respect to Ground . . . . . -0.5V to +7V  
 Power Dissipation . . . . . 1 Watt

## \*COMMENT

Stresses above those listed under "Absolute Maximum Ratings" may cause permanent damage to the device. This is a stress rating only and functional operation of the device at these or any other conditions above those indicated in the operational sections of this specification is not implied.

## D.C. CHARACTERISTICS

T<sub>A</sub> = 0°C to 70°C V<sub>CC</sub> = 5V ± 5% (8259A-8) V<sub>CC</sub> = 5V ± 10% (8259A)

Symbol	Parameter	Min.	Max.	Units	Test Conditions
V <sub>IL</sub>	Input Low Voltage	.5	.8	V	
V <sub>IH</sub>	Input High Voltage	2.0	V <sub>CC</sub> + .5V	V	
V <sub>OL</sub>	Output Low Voltage		.45	V	I <sub>OL</sub> = 2.2 mA
V <sub>OH</sub>	Output High Voltage	2.4		V	I <sub>OH</sub> = -400 μA
V <sub>OH(INT)</sub>	Interrupt Output High Voltage	3.5 2.4		V V	I <sub>OH</sub> = -100 μA I <sub>OH</sub> = -400 μA
I <sub>LI</sub>	Input Load Current		10	μA	V <sub>IN</sub> = V <sub>CC</sub> to 0V
I <sub>LOL</sub>	Output Leakage Current		-10	μA	V <sub>OUT</sub> = 0.45V
I <sub>LOH</sub>	Output Leakage Current		10	μA	V <sub>OUT</sub> = V <sub>CC</sub>
I <sub>CC</sub>	V <sub>CC</sub> Supply Current		85	mA	

## 8259A A.C. CHARACTERISTICS

 $T_A = 0^\circ\text{C}$  to  $70^\circ\text{C}$   $V_{CC} = 5V \pm 5\%$  (8259A-8)  $V_{CC} = 5V \pm 10\%$  (8259A)

## TIMING REQUIREMENTS

8259A-8

8259A

Symbol	Parameter	Min.	Max.	Min.	Max.	Units	Test Conditions
TAHRL	$A0/\overline{CS}$ Setup to $\overline{RD}/\overline{INTA}\downarrow$	50		0		ns	
TRHAX	$A0/\overline{CS}$ Hold after $\overline{RD}/\overline{INTA}\uparrow$	5		0		ns	
TRLRH	$\overline{RD}$ Pulse Width	420		235		ns	
TAHWL	$A0/\overline{CS}$ Setup to $\overline{WR}\downarrow$	50		0		ns	
TWHAX	$A0/\overline{CS}$ Hold after $\overline{WR}\uparrow$	20		0		ns	
TWLWH	$\overline{WR}$ Pulse Width	400		290		ns	
TDVWH	Data Setup to $\overline{WR}\uparrow$	300		240		ns	
TWHDX	Data Hold after $\overline{WR}\uparrow$	40		0		ns	
TJLJH	Interrupt Request Width (Low)	100		100		ns	See Note 1
TCVIAL	Cascade Setup to Second or Third $\overline{INTA}\downarrow$ (Slave Only)	55		55		ns	
TRHRL	End of $\overline{RD}$ to Next Command	160		160		ns	
TWHRL	End of $\overline{WR}$ to Next Command	190		190		ns	

Note: 1. This is the low time required to clear the input latch in the edge triggered mode.

## TIMING RESPONSES

8259A-8

8259A

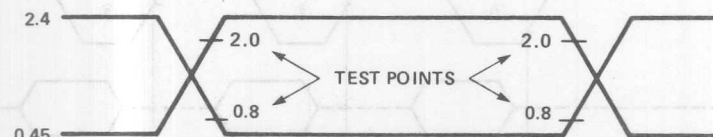
Symbol	Parameter	Min.	Max.	Min.	Max.	Units	Test Conditions
TRLDV	Data Valid from $\overline{RD}/\overline{INTA}\downarrow$		300		200	ns	C of Data Bus Max. test C = 100 pF Min. test C = 15 pF $C_{INT} = 100$ pF $C_{ENABLE} = 15$ pF
TRHDZ	Data Float after $\overline{RD}/\overline{INTA}\uparrow$	10	200		100	ns	
TJHIH	Interrupt Output Delay		400		350	ns	
TIALCV	Cascade Valid from First $\overline{INTA}\downarrow$ (Master Only)		565		565	ns	
TRLEL	Enable Active from $\overline{RD}\downarrow$ or $\overline{INTA}\downarrow$		160		125	ns	
TRHEH	Enable Inactive from $\overline{RD}\uparrow$ or $\overline{INTA}\uparrow$		325		150	ns	
TAHDV	Data Valid from Stable Address		350		200	ns	
TCVDV	Cascade Valid to Valid Data		300		300	ns	

## CAPACITANCE

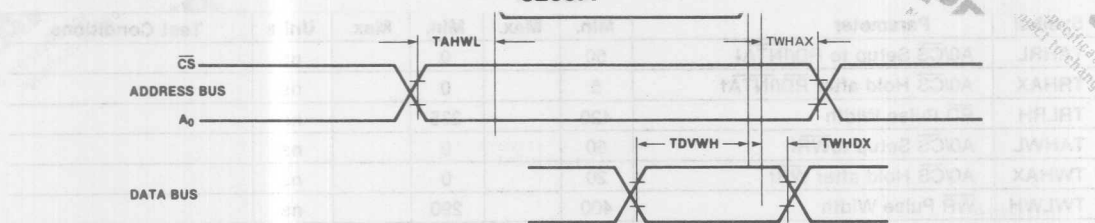
 $T_A = 25^\circ\text{C}$ ;  $V_{CC} = \text{GND} = 0V$ 

Symbol	Parameter	Min.	Typ.	Max.	Unit	Test Conditions
$C_{IN}$	Input Capacitance			10	pF	$f_c = 1$ MHz
$C_{I/O}$	I/O Capacitance			20	pF	Unmeasured pins returned to $V_{SS}$

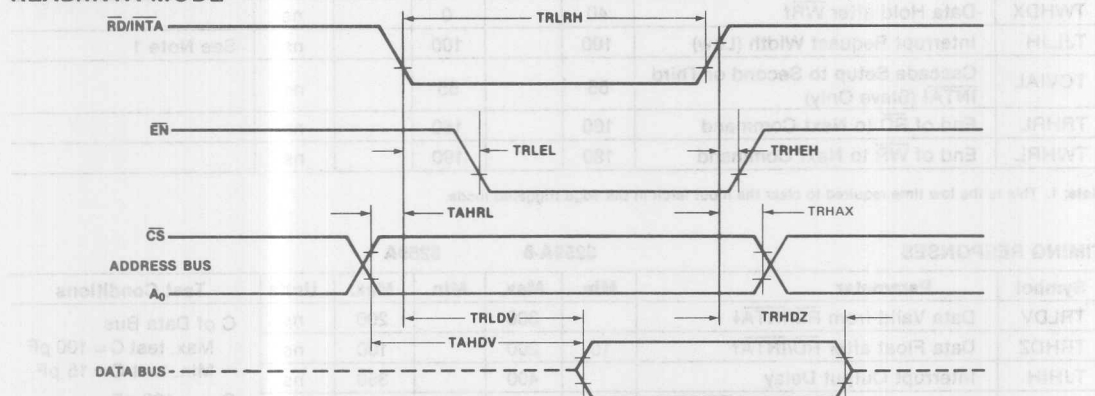
## Input and Output Waveforms for A.C. Tests



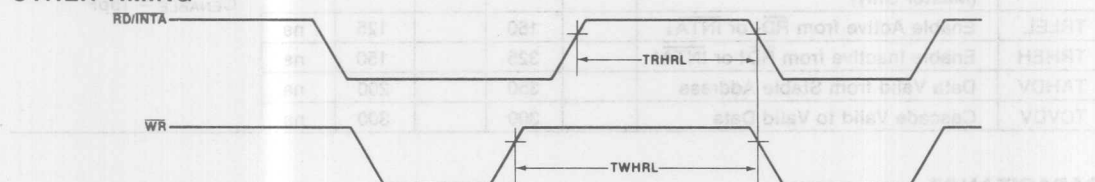
# 8259A



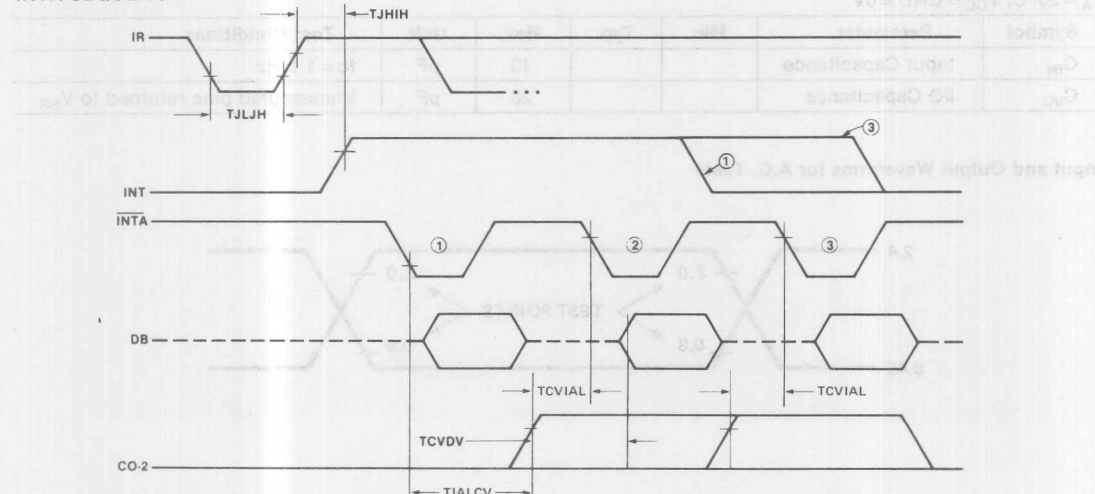
## READ/INTA MODE



## OTHER TIMING



## INTA SEQUENCE



NOTE: Interrupt output must remain "HIGH" (at least) until leading edge of first INTA.

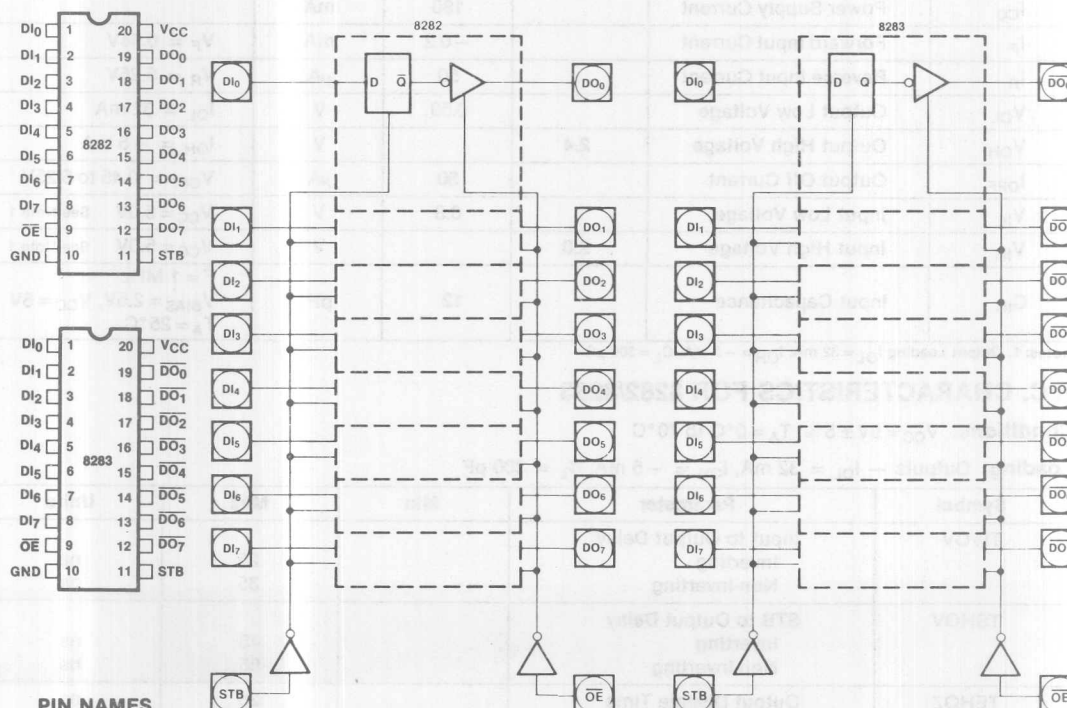
① MCS 8085 Systems only.  
② Cycle 1 in MCS 86 Systems, the Data Bus is not active.

- Fully Parallel 8-Bit Data Register and Buffer
- Transparent during Active Strobe
- Supports 8080, 8085, 8048, and 8086 Systems
- High Output Drive Capability for Driving System Data Bus
- 3-State Outputs
- 20-Pin Package with 0.3" Center
- No Output Low Noise when Entering or Leaving High Impedance State

The 8282 and 8283 are 8-bit bipolar latches with 3-state output buffers. They can be used to implement latches, buffers, or multiplexers. The 8283 inverts the input data at its outputs while the 8282 does not. Thus, all of the principal peripheral and input/output functions of a microcomputer system can be implemented with these devices.

## PIN CONFIGURATIONS

## LOGIC DIAGRAMS



## PIN NAMES

DI <sub>0</sub> -DI <sub>7</sub>	DATA IN
DO <sub>0</sub> -DO <sub>7</sub>	DATA OUT
OE	OUTPUT ENABLE
STB	STROBE

## PIN DEFINITIONS

Pin	Description
STB	STROBE (Input). STB is an input control pulse used to strobe data at the data input pins (A <sub>0</sub> -A <sub>7</sub> ) into the data latches. This signal is active HIGH to admit input data. The data is latched at the HIGH to LOW transition of STB.
$\overline{OE}$	OUTPUT ENABLE (Input). $\overline{OE}$ is an input control signal which when active LOW enables the contents of the data latches onto the data output pin (B <sub>0</sub> -B <sub>7</sub> ). OE being inactive HIGH forces the output buffers to their high impedance state.
DI <sub>0</sub> -DI <sub>7</sub>	DATA INPUT PINS (Input). Data presented at these pins satisfying setup time requirements when STB is strobed and latched into the data input latches.

DO<sub>0</sub>-DO<sub>7</sub>  
(8282)  
 $\overline{DO_0}$ - $\overline{DO_7}$   
(8283)

DATA OUTPUT PINS (Output). When  $\overline{CS}$  is true, the data in the data latches is presented as inverted (8283) or non-inverted (8282) data onto the data output pins.

## OPERATIONAL DESCRIPTION

The 8282 and 8283 octal latches are 8-bit latches with 3-state output buffers. Data having satisfied the setup time requirements is latched into the data latches by strobing the STB line HIGH to LOW. Holding the STB line in its active HIGH state makes the latches appear transparent. Data is presented to the data output pins by activating the  $\overline{OE}$  input line. When  $\overline{OE}$  is inactive HIGH the output buffers are in their high impedance state. Enabling or disabling the output buffers will not cause negative-going transients to appear on the data output bus.

## D.C. CHARACTERISTICS FOR 8282/8283

Conditions:  $V_{CC} = 5V \pm 5\%$ ,  $T_A = 0^\circ C$  to  $70^\circ C$

Symbol	Parameter	Min	Max	Units	Test Conditions
$V_C$	Input Clamp Voltage		-1	V	$I_C = -5 \text{ mA}$
$I_{CC}$	Power Supply Current		160	mA	
$I_F$	Forward Input Current		-0.2	mA	$V_F = 0.45V$
$I_R$	Reverse Input Current		50	$\mu A$	$V_R = 5.25V$
$V_{OL}$	Output Low Voltage		0.50	V	$I_{OL} = 32 \text{ mA}$
$V_{OH}$	Output High Voltage	2.4		V	$I_{OH} = -5 \text{ mA}$
$I_{OFF}$	Output Off Current		50	$\mu A$	$V_{OFF} = 0.45 \text{ to } 5.25V$
$V_{IL}$	Input Low Voltage		0.8	V	$V_{CC} = 5.0V$ See Note 1
$V_{IH}$	Input High Voltage	2.0		V	$V_{CC} = 5.0V$ See Note 1
$C_{IN}$	Input Capacitance		12	pF	$F = 1 \text{ MHz}$ $V_{BIAS} = 2.5V$ , $V_{CC} = 5V$ $T_A = 25^\circ C$

Notes: 1. Output Loading  $I_{OL} = 32 \text{ mA}$ ,  $I_{OH} = -5 \text{ mA}$ ,  $C_L = 300 \text{ pF}$

## A.C. CHARACTERISTICS FOR 8282/8283

Conditions:  $V_{CC} = 5V \pm 5\%$ ,  $T_A = 0^\circ C$  to  $70^\circ C$

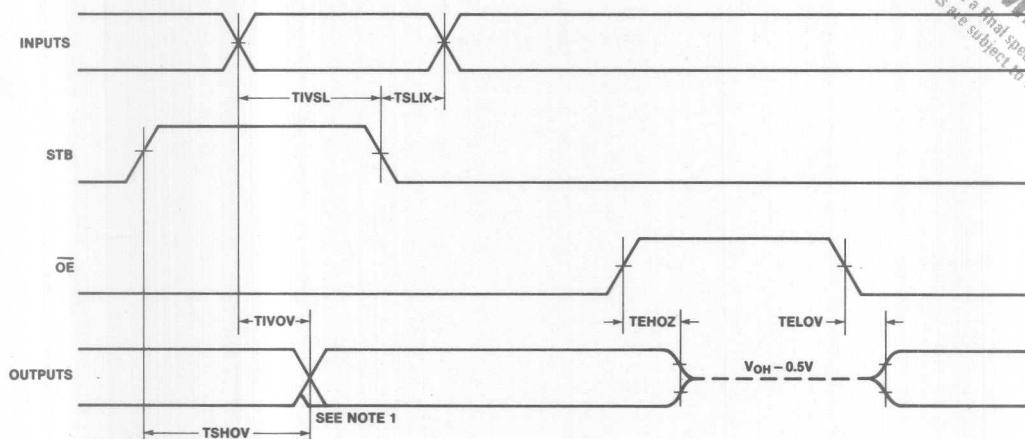
Loading: Outputs —  $I_{OL} = 32 \text{ mA}$ ,  $I_{OH} = -5 \text{ mA}$ ,  $C_L = 300 \text{ pF}$

Symbol	Parameter	Min	Max	Units
T <sub>IOV</sub>	Input to Output Delay Inverting		25	ns
			35	ns
T <sub>SHOV</sub>	STB to Output Delay Inverting		45	ns
			55	ns
TEHOZ	Output Disable Time		25	ns
TELOV	Output Enable Time	10	50	ns
T <sub>IVSL</sub>	Input to STB Setup Time	0		ns
T <sub>SLIX</sub>	Input to STB Hold Time	25		ns

Notes: 1. See waveforms and test load circuit on following page.

## 8282/8283 TIMING

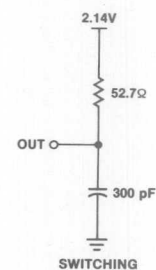
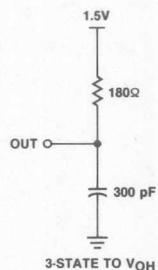
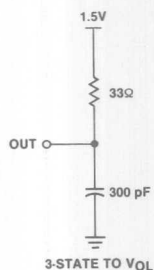
**PRELIMINARY**  
 Notice: This is not a final specification. Some  
 parametric limits are subject to change.



**NOTE:** 1. 8283 ONLY — OUTPUT MAY BE MOMENTARILY INVALID FOLLOWING THE HIGH GOING STB TRANSITION.

2. ALL TIMING MEASUREMENTS ARE MADE AT 1.5V UNLESS OTHERWISE NOTED

## OUTPUT TEST LOAD CIRCUITS





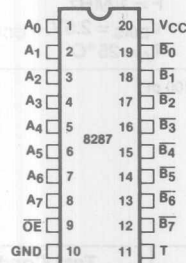
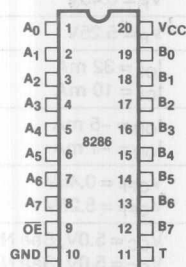
# 8286/8287

## 8-BIT PARALLEL BIDIRECTIONAL BUS DRIVERS

- Data Bus Buffer Driver for MCS-86™, MCS-80™, MCS-85™, and MCS-48™
- High Output Drive Capability for Driving System Data Bus
- Fully Parallel 8-Bit Transceivers
- 3-State Outputs
- 20-Pin Package with 0.3" Center
- No Output Low Noise when Entering or Leaving High Impedance State

The 8286 and 8287 are 8-bit bipolar transceivers with 3-state outputs. The 8287 inverts the input data at its outputs while the 8286 does not. Thus, a wide variety of applications for buffering in microcomputer systems can be met.

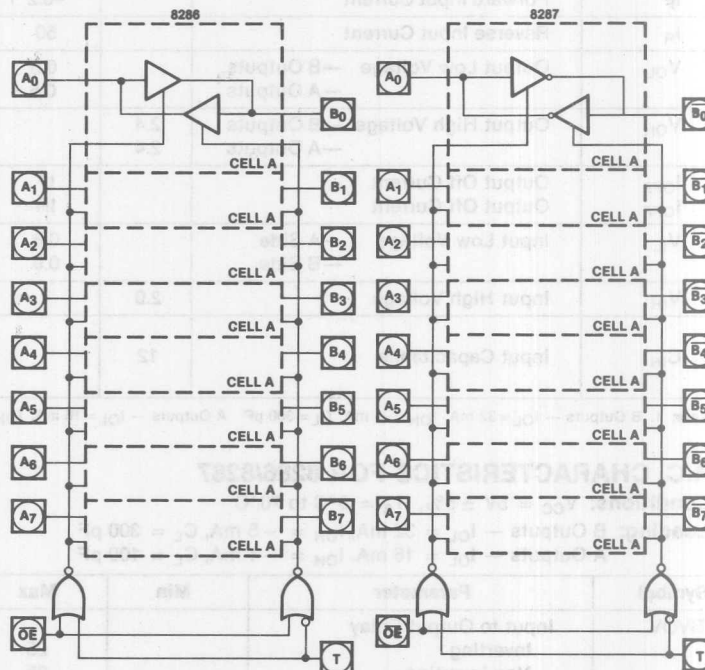
### PIN CONFIGURATIONS



### PIN NAMES

A <sub>0</sub> -A <sub>7</sub>	LOCAL BUS DATA
B <sub>0</sub> -B <sub>7</sub>	SYSTEM BUS DATA
OE	OUTPUT ENABLE
T	TRANSMIT

### LOGIC DIAGRAMS



## PIN DEFINITIONS

Pin	Description
T	TRANSMIT (Input). T is an input control signal used to control the direction of the transceivers. When HIGH, it configures the transceiver's B <sub>0</sub> -B <sub>7</sub> as outputs with A <sub>0</sub> -A <sub>7</sub> as inputs. T LOW configures A <sub>0</sub> -A <sub>7</sub> as the outputs with B <sub>0</sub> -B <sub>7</sub> serving as the inputs.
$\overline{OE}$	OUTPUT ENABLE (Input). $\overline{OE}$ is an input control signal used to enable the appropriate output driver (as selected by T) onto its respective bus. This signal is active LOW.
A <sub>0</sub> -A <sub>7</sub>	LOCAL BUS DATA PINS (Input/Output). These pins serve to either present data to or accept data from the processor's local bus depending upon the state of the T pin.

B<sub>0</sub>-B<sub>7</sub>  
(8286)  
B<sub>0</sub>-B<sub>7</sub>  
(8287)

SYSTEM BUS DATA PINS (Input/Output). These pins serve to either present data to or accept data from the system bus depending upon the state of the T pin.

## OPERATIONAL DESCRIPTION

The 8286 and 8287 transceivers are 8-bit transceivers with high impedance outputs. With T active HIGH and  $\overline{OE}$  active LOW, data at the A<sub>0</sub>-A<sub>7</sub> pins is driven onto the B<sub>0</sub>-B<sub>7</sub> pins. With T inactive LOW and  $\overline{OE}$  active LOW, data at the B<sub>0</sub>-B<sub>7</sub> pins is driven onto the A<sub>0</sub>-A<sub>7</sub> pins. No output low glitching will occur whenever the transceivers are entering or leaving the high impedance state.

## D.C. CHARACTERISTICS FOR 8286/8287

Conditions: V<sub>CC</sub> = 5V ± 5%, T<sub>A</sub> = 0°C to 70°C

Symbol	Parameter	Min	Max	Units	Test Conditions
V <sub>C</sub>	Input Clamp Voltage		-1	V	I <sub>C</sub> = -5 mA
I <sub>CC</sub>	Power Supply Current—8287 —8286		95 135	mA mA	
I <sub>F</sub>	Forward Input Current		-0.2	mA	V <sub>F</sub> = 0.45V
I <sub>R</sub>	Reverse Input Current		50	μA	V <sub>R</sub> = 5.25V
V <sub>OL</sub>	Output Low Voltage —B Outputs —A Outputs		0.5 0.5	V V	I <sub>OL</sub> = 32 mA I <sub>OL</sub> = 10 mA
V <sub>OH</sub>	Output High Voltage —B Outputs —A Outputs	2.4 2.4		V V	I <sub>OH</sub> = -5 mA I <sub>OH</sub> = -1 mA
I <sub>OFF</sub> I <sub>OFF</sub>	Output Off Current Output Off Current		I <sub>F</sub> I <sub>R</sub>		V <sub>OFF</sub> = 0.45V V <sub>OFF</sub> = 5.25V
V <sub>IL</sub>	Input Low Voltage —A Side —B Side		0.8 0.9	V V	V <sub>CC</sub> = 5.0V, See Note 1 V <sub>CC</sub> = 5.0V, See Note 1
V <sub>IH</sub>	Input High Voltage	2.0		V	V <sub>CC</sub> = 5.0V, See Note 1
C <sub>IN</sub>	Input Capacitance	12		pF	F = 1 MHz V <sub>BIAS</sub> = 2.5V, V <sub>CC</sub> = 5V T <sub>A</sub> = 25°C

Note: 1. B Outputs — I<sub>OL</sub> = 32 mA, I<sub>OH</sub> = -5 mA, C<sub>L</sub> = 300 pF A Outputs — I<sub>OL</sub> = 16 mA, I<sub>OH</sub> = -1 mA, C<sub>L</sub> = 100 pF

## A.C. CHARACTERISTICS FOR 8286/8287

Conditions: V<sub>CC</sub> = 5V ± 5%, T<sub>A</sub> = 0°C to 70°C

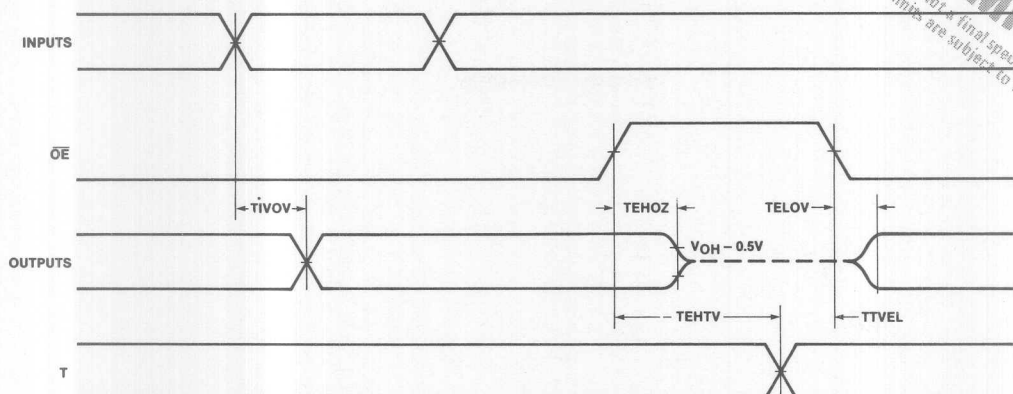
Loading: B Outputs — I<sub>OL</sub> = 32 mA, I<sub>OH</sub> = -5 mA, C<sub>L</sub> = 300 pF  
A Outputs — I<sub>OL</sub> = 16 mA, I<sub>OH</sub> = -1 mA, C<sub>L</sub> = 100 pF

Symbol	Parameter	Min	Max	Units	Test Conditions
T <sub>IOV</sub>	Input to Output Delay Inverting Non-Inverting		25 35	ns ns	(See Note 1)
T <sub>HTV</sub>	Transmit/Receive Hold Time	TEHOZ		ns	
T <sub>TVEL</sub>	Transmit/Receive Setup	30		ns	
TEHOZ	Output Disable Time		25	ns	
TELOV	Output Enable Time	10	50	ns	

Note: 1. See waveforms and test load circuit on following page.

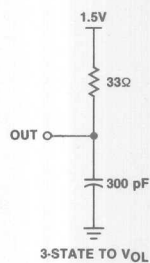
## 8286/8287 TIMING

**PRELIMINARY**  
 Notice: This is not a final specification. Some parametric limits are subject to change.

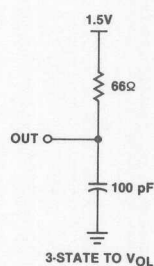


NOTE: 1. ALL TIMING MEASUREMENTS ARE MADE AT 1.5V UNLESS OTHERWISE NOTED.

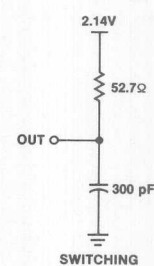
## TEST LOAD CIRCUITS



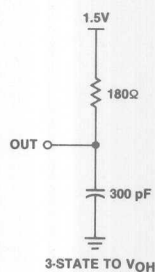
B OUTPUT



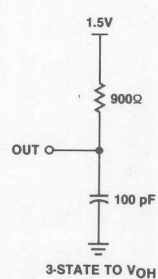
A OUTPUT



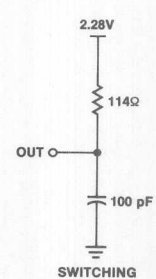
B OUTPUT



B OUTPUT



A OUTPUT



A OUTPUT



## CHAPTER 5

**MCS-85**

**System Support  
Components**

**Peripherals**

**Static RAMs**

**ROMs/EPROMs**

**MCS  
85**

**MCS  
85**

**MCS  
85**





# 8041/8741

## UNIVERSAL PERIPHERAL INTERFACE

### 8-BIT MICROCOMPUTER

**PRELIMINARY**  
Notice: This is not a final specification. Some parametric limits are subject to change.

- Fully Compatible with MCS-80™, MCS-85™ and MCS-48™ Microprocessor Families
- Single Level Interrupt
- 8-Bit CPU plus ROM, RAM, I/O, Timer and Clock in a Single Package
- Single 5V Supply
- Alternative to Custom LSI
- Pin Compatible ROM and EPROM Versions
- 1K x 8 ROM/EPROM, 64 x 8 RAM, 18 Programmable I/O Pins
- Asynchronous Data Register for Interface to Master Processor
- Expandable I/O

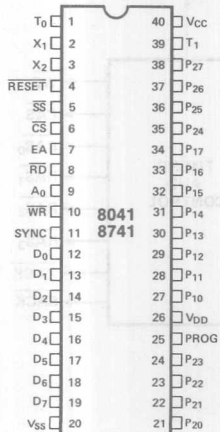
The Intel® 8041/8741 is a general purpose, programmable interface device designed for use with a variety of 8-bit microprocessor systems. It contains a low cost microcomputer with program memory, data memory, 8-bit CPU, I/O ports, timer/counter, and clock in a single 40-pin package. Interface registers are included to enable the UPI device to function as a peripheral controller in MCS-80™, MCS-85™, MCS-48™, and other 8-bit systems.

The UPI-41™ has 1K words of program memory and 64 words of data memory on-chip. To allow full user flexibility the program memory is available as ROM in the 8041 version or as UV-erasable EPROM in the 8741 version. The 8741 and the 8041 are fully pin compatible for easy transition from prototype to production level designs.

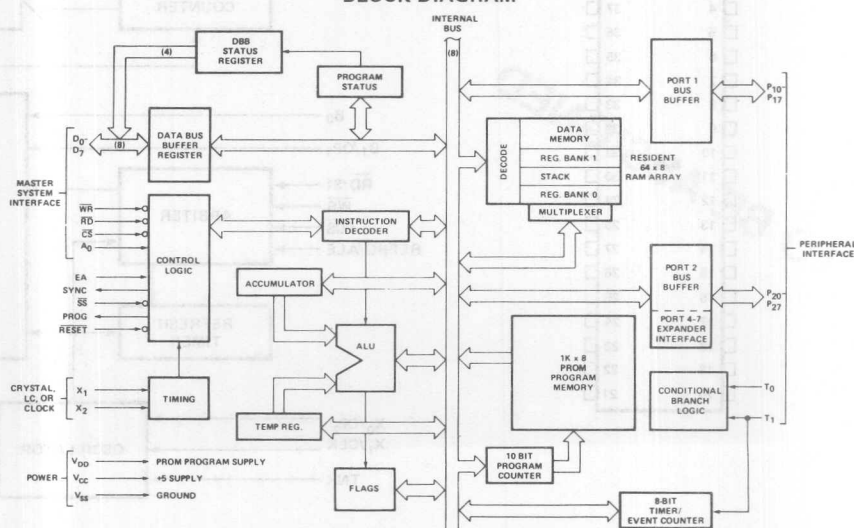
The device has two 8-bit, TTL compatible I/O ports and two test inputs. Individual port lines can function as either inputs or outputs under software control. I/O can be expanded with the 8243 device which is directly compatible and has 16 I/O lines. An 8-bit programmable timer/counter is included in the UPI device for generating timing sequences or counting external inputs. Additional UPI features include: single 5V supply, low power standby mode (in the 8041), single-step mode for debug (in the 8741), single level interrupt, and dual working register banks.

Because it's a complete microcomputer, the UPI provides more flexibility for the designer than conventional LSI interface devices. It is designed to be an efficient controller as well as an arithmetic processor. Applications include keyboard scanning, printer control, display multiplexing and similar functions which involve interfacing peripheral devices to microprocessor systems.

#### PIN CONFIGURATION



#### BLOCK DIAGRAM

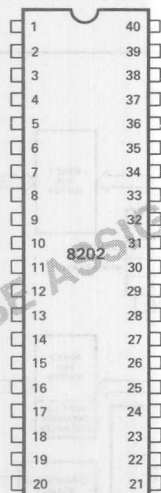


# 8202 DYNAMIC RAM CONTROLLER

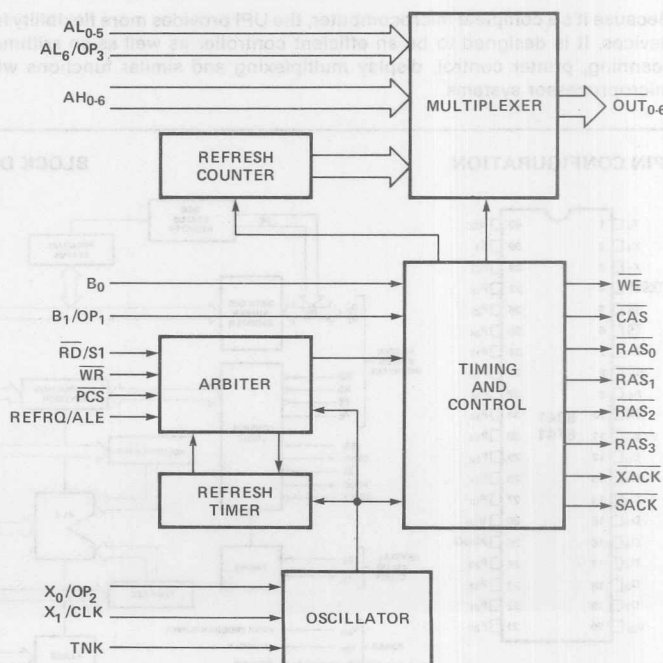
- Provides All Signals Necessary to Control 2104A, 2116, or 2117 Dynamic Memories
- Directly Addresses and Drives Up to 64K Bytes Without External Drivers
- Provides Address Multiplexing and Strokes
- Provides a Refresh Timer and a Refresh Counter
- Refresh Cycles May be Internally or Externally Requested
- Provides High Speed Refresh/Memory Access Request Arbitration
- Provides Transparent Refresh Capability
- Fully Compatible with Intel® 8080A and 8085A Microprocessors
- Decodes 8085A Status for Advanced Read Capability
- Provides System Acknowledge and Transfer Acknowledge Signals
- Internal or External Clock Capability

The 8202 is a Dynamic RAM System Controller designed to provide all signals necessary to use 2104A, 2116, or 2117 Dynamic RAMs in microcomputer systems. The 8202 provides multiplexed addresses and address strobes, as well as refresh/access arbitration. Refresh cycles can be started internally or externally.

PIN CONFIGURATION



8202 BLOCK DIAGRAM



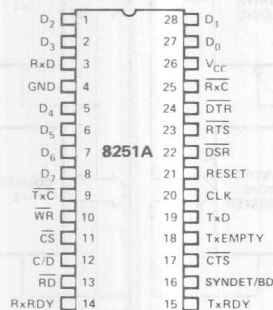


## 8251A PROGRAMMABLE COMMUNICATION INTERFACE

- Synchronous and Asynchronous Operation
- Synchronous 5-8 Bit Characters; Internal or External Character Synchronization; Automatic Sync Insertion
- Asynchronous 5-8 Bit Characters; Clock Rate—1, 16 or 64 Times Baud Rate; Break Character Generation; 1, 1½, or 2 Stop Bits; False Start Bit Detection; Automatic Break Detect and Handling; 19.2K Baud.
- Baud Rate — DC to 64K Baud
- Full Duplex, Double Buffered, Transmitter and Receiver
- Error Detection — Parity, Overrun and Framing
- Fully Compatible with 8080/8085 CPU
- 28-Pin DIP Package
- All Inputs and Outputs are TTL Compatible
- Single +5V Supply
- Single TTL Clock

The Intel® 8251A is the enhanced version of the industry standard, Intel® 8251 Universal Synchronous/Asynchronous Receiver/Transmitter (USART), designed for data communications with Intel's new high performance family of microprocessors such as the 8085. The 8251A is used as a peripheral device and is programmed by the CPU to operate using virtually any serial data transmission technique presently in use (including IBM "bi-sync"). The USART accepts data characters from the CPU in parallel format and then converts them into a continuous serial data stream for transmission. Simultaneously, it can receive serial data streams and convert them into parallel data characters for the CPU. The USART will signal the CPU whenever it can accept a new character for transmission or whenever it has received a character for the CPU. The CPU can read the complete status of the USART at any time. These include data transmission errors and control signals such as SYNDet, TxEMPTY. The chip is constructed using N-channel silicon gate technology.

PIN CONFIGURATION

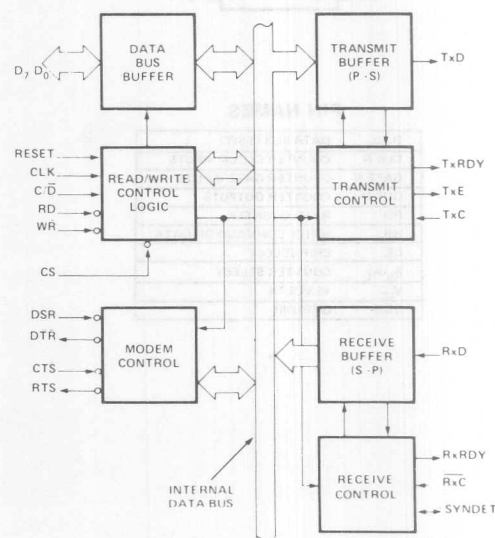


PIN NAMES

D <sub>7</sub> D <sub>0</sub>	Data Bus (8 bits)
C/D	Control or Data is to be Written or Read
RD	Read Data Command
WR	Write Data or Control Command
CS	Chip Enable
CLK	Clock Pulse (TTL)
RESET	Reset
TxC	Transmitter Clock
TxD	Transmitter Data
RxC	Receiver Clock
RxD	Receiver Data
RxRDY	Receiver Ready (has character for 8080)
TxRDY	Transmitter Ready (ready for char. from 8080)

DSR	Data Set Ready
DTR	Data Terminal Ready
SYNDet/BD	Sync Detect/ Break Detect
RTS	Request to Send Data
CTS	Clear to Send Data
TxE	Transmitter Empty
V <sub>CC</sub>	+5 Volt Supply
GND	Ground

BLOCK DIAGRAM



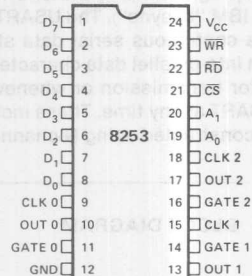
# 8253/8253-5 PROGRAMMABLE INTERVAL TIMER

- MCS—85™ Compatible 8253-5
- 3 Independent 16-Bit Counters
- DC to 2 MHz
- Programmable Counter Modes
- Count Binary or BCD
- Single +5V Supply
- 24-Pin Dual In-Line Package

The Intel® 8253 is a programmable counter/timer chip designed for use as an Intel microcomputer peripheral. It uses nMOS technology with a single +5V supply and is packaged in a 24-pin plastic DIP.

It is organized as 3 independent 16-bit counters, each with a count rate of up to 2 MHz. All modes of operation are software programmable.

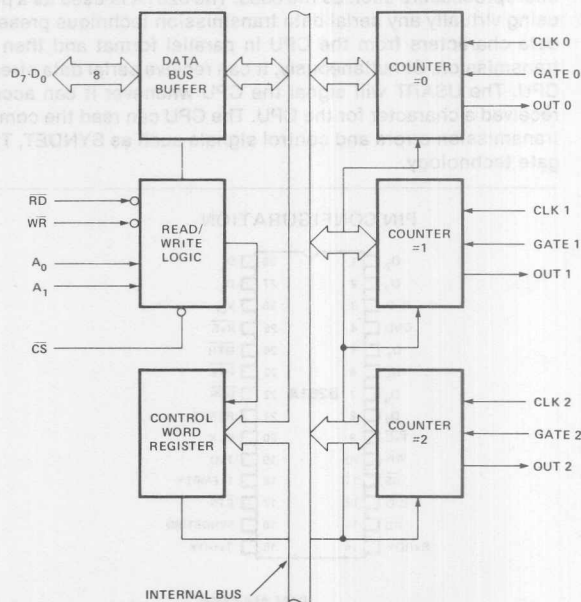
## PIN CONFIGURATION



## PIN NAMES

D <sub>7</sub> -D <sub>0</sub>	DATA BUS (8-BIT)
CLK N	COUNTER CLOCK INPUTS
GATE N	COUNTER GATE INPUTS
OUT N	COUNTER OUTPUTS
RD	READ COUNTER
WR	WRITE COMMAND OR DATA
CS	CHIP SELECT
A <sub>0</sub> A <sub>1</sub>	COUNTER SELECT
V <sub>CC</sub>	+5 VOLTS
GND	GROUND

## BLOCK DIAGRAM





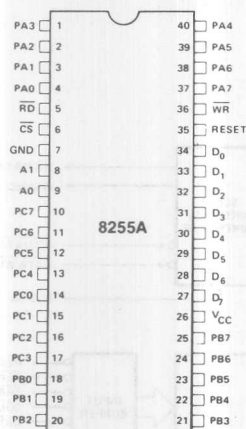
**PRELIMINARY**  
Notice: This is not a final specification. Some parametric limits are subject to change.

## 8255A/8255A-5 PROGRAMMABLE PERIPHERAL INTERFACE

- MCS-85™ Compatible 8255A-5
- 24 Programmable I/O Pins
- Completely TTL Compatible
- Fully Compatible with Intel® Micro-processor Families
- Improved Timing Characteristics
- Direct Bit Set/Reset Capability Easing Control Application Interface
- 40-Pin Dual In-Line Package
- Reduces System Package Count
- Improved DC Driving Capability

The Intel® 8255A is a general purpose programmable I/O device designed for use with Intel® microprocessors. It has 24 I/O pins which may be individually programmed in 2 groups of 12 and used in 3 major modes of operation. In the first mode (MODE 0), each group of 12 I/O pins may be programmed in sets of 4 to be input or output. In MODE 1, the second mode, each group may be programmed to have 8 lines of input or output. Of the remaining 4 pins, 3 are used for handshaking and interrupt control signals. The third mode of operation (MODE 2) is a bidirectional bus mode which uses 8 lines for a bidirectional bus, and 5 lines, borrowing one from the other group, for handshaking.

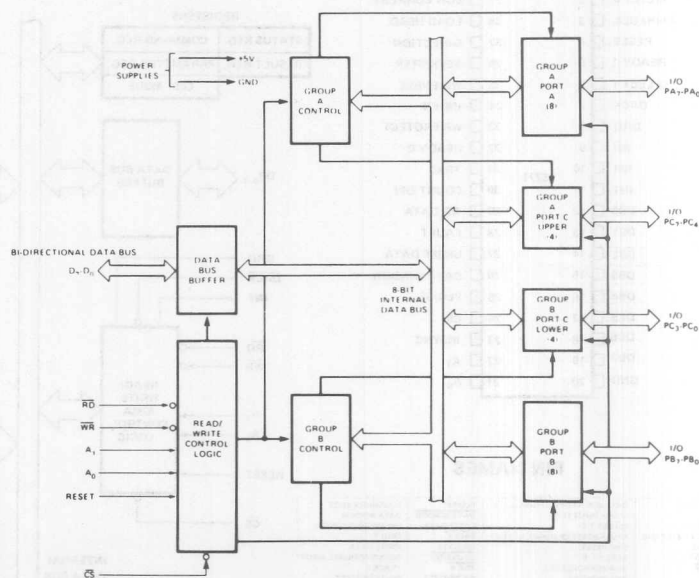
PIN CONFIGURATION



PIN NAMES

D <sub>7</sub> -D <sub>0</sub>	DATA BUS (BI-DIRECTIONAL)
RESET	RESET INPUT
CS	CHIP SELECT
RD	READ INPUT
WR	WRITE INPUT
A <sub>0</sub> , A <sub>1</sub>	PORT ADDRESS
PA <sub>7</sub> -PA <sub>0</sub>	PORT A (BIT)
PB <sub>7</sub> -PB <sub>0</sub>	PORT B (BIT)
PC <sub>7</sub> -PC <sub>0</sub>	PORT C (BIT)
V <sub>CC</sub>	+5 VOLTS
GND	0 VOLTS

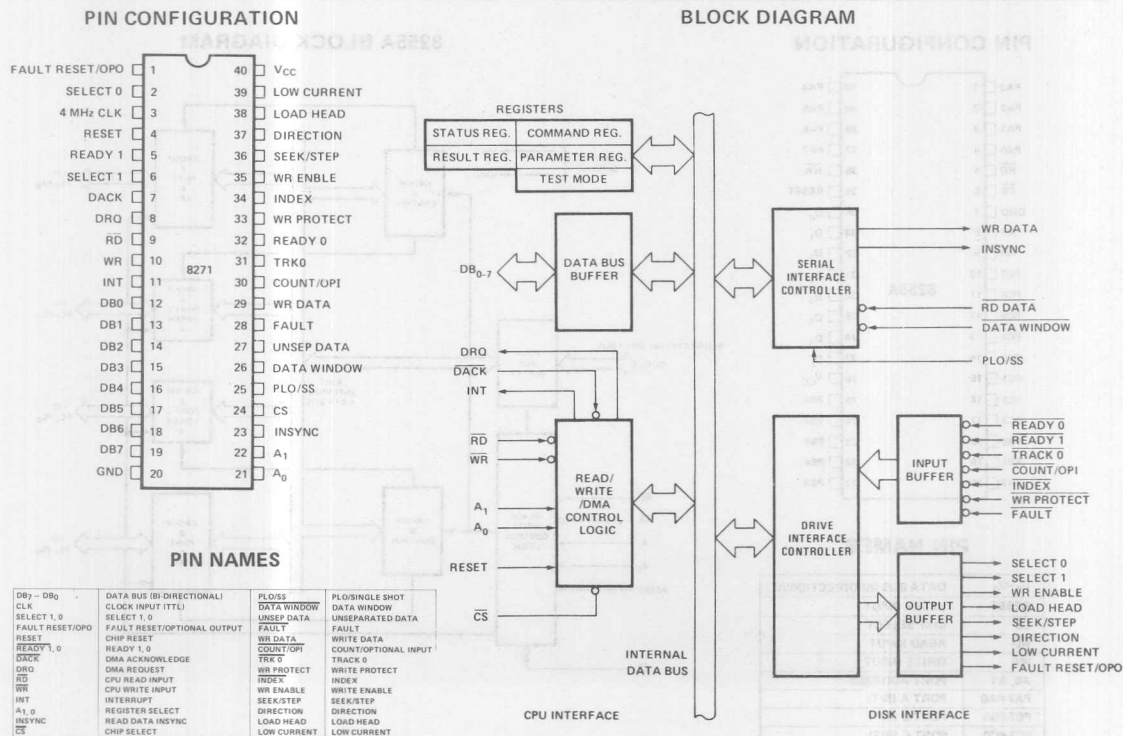
8255A BLOCK DIAGRAM



# 8271 PROGRAMMABLE FLOPPY DISK CONTROLLER

- IBM 3740 Soft Sector Format Compatible
- Programmable Record Lengths
- Multi-Sector Capability
- Maintain Dual Drives with Minimum Software Overhead Expandable to 4 Drives
- Automatic Read/Write Head Positioning and Verification
- Internal CRC Generation and Checking
- Programmable Step Rate, Settle-Time, Head Load Time, Head Unload Index Count
- Fully MCS-80 and MCS-85 Compatible
- Single +5V Supply
- 40-Pin Package

The Intel® 8271 Programmable Floppy Disk Controller (FDC) is an LSI component designed to interface one to 4 floppy disk drives to an 8-bit microcomputer system. Its powerful control functions minimize both hardware and software overhead normally associated with floppy disk controllers.

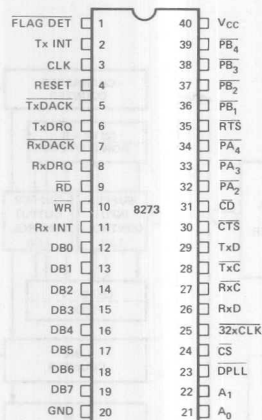


# 8273 PROGRAMMABLE HDLC/SDLC PROTOCOL CONTROLLER

- HDLC/SDLC Compatible
- Frame Level Commands
- Full Duplex, Half Duplex, or Loop SDLC Operation
- Up to 64K Baud Transfers
- Two User Programmable Modem Control Ports
- Automatic FCS (CRC) Generation and Checking
- Programmable NRZI Encode/Decode
- N-Bit Reception Capability
- Digital Phase Locked Loop Clock Recovery
- Minimum CPU Overhead
- Fully Compatible with 8080/8085 CPUs
- Single +5V Supply
- 40-Pin Package

The Intel® 8273 Programmable HDLC/SDLC Protocol Controller is a dedicated device designed to support the ISO/CITT's HDLC and IBM's SDLC communication line protocols. It is fully compatible with Intel's new high performance microcomputer systems such as the MCS-85™. A frame level command set is achieved by a unique microprogrammed dual processor chip architecture. The processing capability supported by the 8273 relieves the system CPU of the low level real-time tasks normally associated with controllers.

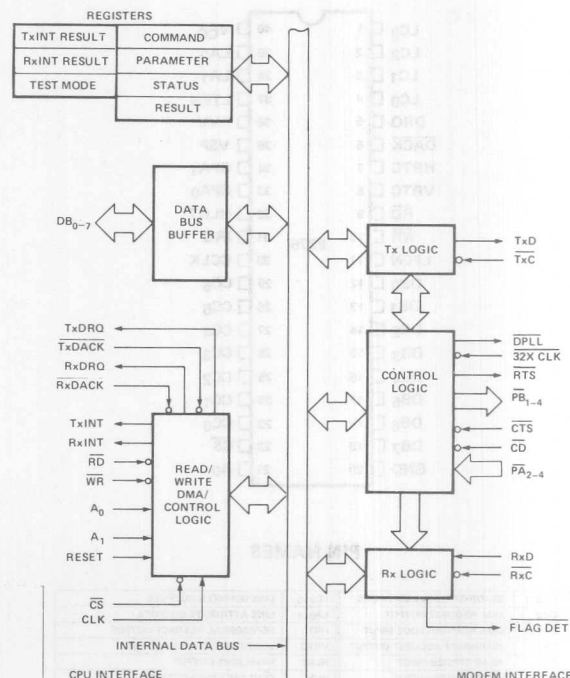
## PIN CONFIGURATION



## PIN NAMES

DB0-DB7	DATA BUS (8 BITS)	CS	CHIP SELECT
FLAG DET	FLAG DETECT	32xCLK	32 TIMES CLOCK
Tx INT	TRANSMITTER INTERRUPT	Rx D	RECEIVER DATA
CLK	CLOCK INPUT	Rx C	RECEIVER CLOCK
RESET	RESET	Tx C	TRANSMITTER CLOCK
Tx DACK	TRANSMITTER DMA ACKNOWLEDGE	Tx D	TRANSMITTER DATA
Tx DRQ	TRANSMITTER DMA REQUEST	CTS	CLEAR TO SEND
RD	READ INPUT	CD	CARRIER DETECT
WR	WRITE INPUT	GP	GP INPUT PORTS
Rx DACK	RECEIVER DMA ACKNOWLEDGE	PA2-PA4	GP OUTPUT PORTS
Rx DRQ	RECEIVER DMA REQUEST	PB1-PB4	REQUEST TO SEND
Rx INT	RECEIVER INTERRUPT	RTS	+5 VOLT SUPPLY
A0-A1	COMMAND REGISTER SELECT ADDRESS	VCC	GROUND
DPLL	DIGITAL PHASE LOCKED LOOP	GND	

## BLOCK DIAGRAM

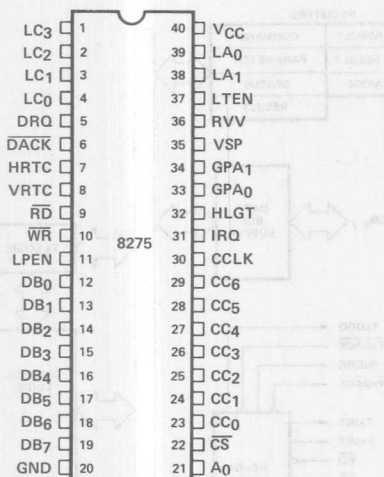


# 8275 PROGRAMMABLE CRT CONTROLLER

- Programmable Screen and Character Format
- 6 Independent Visual Field Attributes
- 11 Visual Character Attributes (Graphic Capability)
- Cursor Control (4 Types)
- Light Pen Detection and Registers
- Fully MCS-80™ and MCS-85™ Compatible
- Dual Row Buffers
- Programmable DMA Burst Mode
- Single +5V Supply
- 40-Pin Package

The Intel® 8275 Programmable CRT Controller is a single chip device to interface CRT raster scan displays with Intel® microcomputer systems. Its primary function is to refresh the display by buffering the information from main memory and keeping track of the display position of the screen. The flexibility designed into the 8275 will allow simple interface to almost any raster scan CRT display with a minimum of external hardware and software overhead.

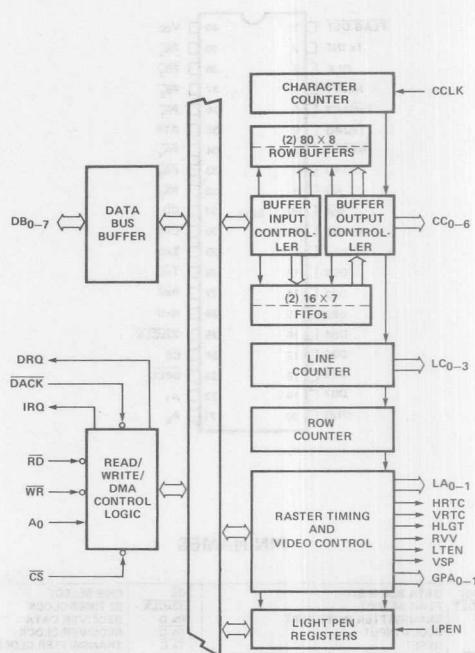
## PIN CONFIGURATION



## PIN NAMES

DB <sub>0-7</sub>	B <sub>1</sub> -DIRECTIONAL DATA BUS	LC <sub>0-3</sub>	LINE COUNTER OUTPUTS
DRQ	DMA REQUEST OUTPUT	LA <sub>0-1</sub>	LINE ATTRIBUTE OUTPUTS
DACK	DMA ACKNOWLEDGE INPUT	HRTC	HORIZONTAL RETRACE OUTPUT
IRQ	INTERRUPT REQUEST OUTPUT	VRTC	VERTICAL RETRACE OUTPUT
RD	READ STROBE INPUT	HLGT	HIGHLIGHT OUTPUT
WR	WRITE STROBE INPUT	RVV	REVERSE VIDEO OUTPUT
A <sub>0</sub>	REGISTER ADDRESS INPUT	LTEN	LIGHT ENABLE OUTPUT
CS	CHIP SELECT INPUT	VSP	VIDEO SUPPRESS OUTPUT
CCLK	CHARACTER CLOCK INPUT	GPA <sub>0-1</sub>	GENERAL PURPOSE ATTRIBUTE OUTPUTS
CC <sub>0-6</sub>	CHARACTER CODE OUTPUTS	LPEN	LIGHT PEN INPUT

## BLOCK DIAGRAM



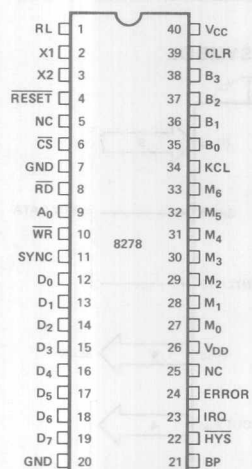
# 8278 PROGRAMMABLE KEYBOARD INTERFACE

- Simultaneous Keyboard and Display Operations
- Interface Signals for Contract and Capacitive Coupled Keyboards
- 128-Key Scanning Logic
- 10.7 msec Matrix Scan Time for 128 Keys and 6 MHz Clock
- 8-Character Keyboard FIFO
- N-Key Rollover with Programmable Error Mode on Multiple New Closures
- 16- or 18-Character 7-Segment Display Interface
- Right or Left Entry Display RAM
- Depress/Release Mode Programmable
- Interrupt Output on Key Entry

The Intel® 8278 is a general purpose programmable keyboard and display interface device designed for use with 8-bit microprocessors such as the MDS-80™ and MCS-85™. The keyboard portion can provide a scanned interface to 128-key contact or capacitive-coupled keyboards. The keys are fully debounced with N-key rollover and programmable error generation on multiple new key closures. Keyboard entries are stored in an 8-character FIFO with overrun status indication when more than 8 characters are entered. Key entries set an interrupt request output to the master CPU.

The display portion of the 8278 provides a scanned display interface for LED, incandescent, and other popular display technologies. Both numeric displays and simple indicators may be used. The 8278 has a 16X4 display RAM which can be loaded or interrogated by the CPU. Both right entry calculator and left entry typewriter display formats are possible. Both read and write of the display RAM can be done with auto-increment of the display RAM address.

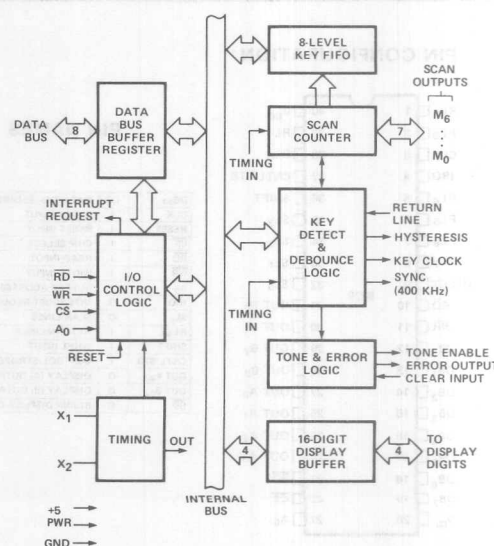
## PIN CONFIGURATION



## PIN NAMES

D <sub>7</sub> -D <sub>0</sub>	DATA BUS
RD, WR	READ, WRITE STROBES
CS	CHIP SELECT
A <sub>0</sub>	CONTROL/DATA SELECT
RESET	RESET INPUT
X <sub>1</sub> , X <sub>2</sub>	FREQ. REFERENCE INPUT
SYNC	HIGH FREQUENCY OUTPUT CLOCK
RL	KEYBOARD RETURN LINE
CLR	CLEAR ERROR
KCL	KEY CLOCK
M <sub>6</sub> -M <sub>0</sub>	MATRIX SCAN LINES
B <sub>3</sub> -B <sub>0</sub>	DISPLAY OUTPUTS
ERROR	ERROR SIGNAL
IRQ	INTERRUPT REQUEST
HYS	HYSTERESIS
BP	tone ENABLE

## BLOCK DIAGRAM



# 8279/8279-5

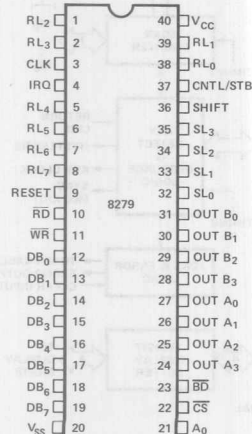
## PROGRAMMABLE KEYBOARD/DISPLAY INTERFACE

- MCS-85™ Compatible 8279-5
- Simultaneous Keyboard Display Operations
- Scanned Keyboard Mode
- Scanned Sensor Mode
- Strobed Input Entry Mode
- 8-Character Keyboard FIFO
- 2-Key Lockout or N-Key Rollover with Contact Debounce
- Dual 8- or 16-Numerical Display
- Single 16-Character Display
- Right or Left Entry 16-Byte Display RAM
- Mode Programmable from CPU
- Programmable Scan Timing
- Interrupt Output on Key Entry

The Intel® 8279 is a general purpose programmable keyboard and display I/O interface device designed for use with Intel® microprocessors. The keyboard portion can provide a scanned interface to a 64-contact key matrix. The keyboard portion will also interface to an array of sensors or a strobed interface keyboard, such as the hall effect and ferrite variety. Key depressions can be 2-key lockout or N-key rollover. Keyboard entries are debounced and strobed in an 8-character FIFO. If more than 8 characters are entered, overrun status is set. Key entries set the interrupt output line to the CPU.

The display portion provides a scanned display interface for LED, incandescent, and other popular display technologies. Both numeric and alphanumeric segment displays may be used as well as simple indicators. The 8279 has 16X8 display RAM which can be organized into dual 16X4. The RAM can be loaded or interrogated by the CPU. Both right entry, calculator and left entry typewriter display formats are possible. Both read and write of the display RAM can be done with auto-increment of the display RAM address.

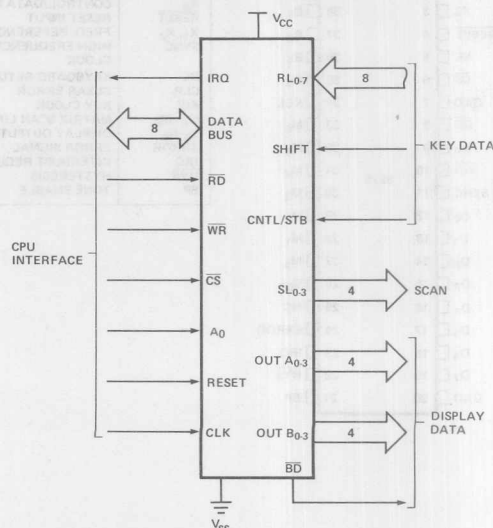
### PIN CONFIGURATION



### PIN NAMES

DB <sub>0-7</sub>	I/O	DATA BUS (BI-DIRECTIONAL)
CLK	I	CLOCK INPUT
RESET	I	RESET INPUT
CS	I	CHIP SELECT
RD	I	READ INPUT
WR	I	WRITE INPUT
A <sub>0</sub>	I	BUFFER ADDRESS
IRQ	O	INTERRUPT REQUEST OUTPUT
SL <sub>0-3</sub>	O	SCAN LINES
RL <sub>0-7</sub>	I	RETURN LINES
SHIFT	I	SHIFT INPUT
CNTL/STB	I	CONTROL/STROBE INPUT
OUT A <sub>0-3</sub>	O	DISPLAY (A) OUTPUTS
OUT B <sub>0-3</sub>	O	DISPLAY (B) OUTPUTS
BD	O	BLANK DISPLAY OUTPUT

### LOGIC SYMBOL



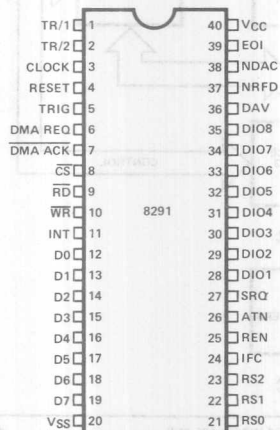
# 8291 GPIB TALKER/LISTENER

**PRELIMINARY**  
 Notice: This is not a final specification. Some parametric limits are subject to change.

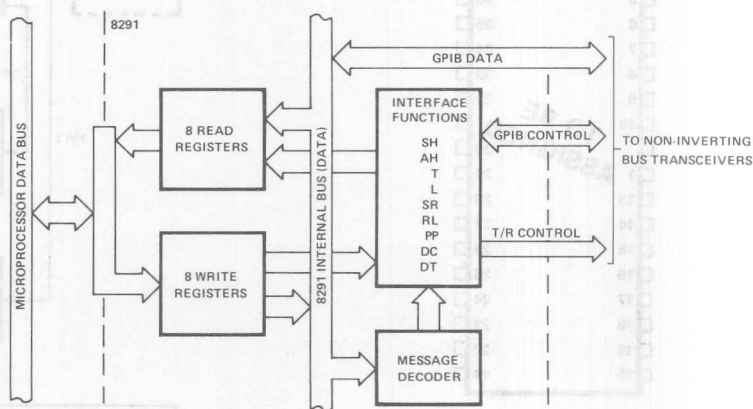
- Designed to Interface Microprocessors (e.g., 8080, 8085, 8086, 8048) to an IEEE Standard 488 Digital Interface Bus
- Programmable Data Transfer Rate
- Complete Source and Acceptor Handshake
- Complete Talker and Listener Functions with Extended Addressing
- Service Request, Parallel Poll, Device Clear, Device Trigger, Remote/Local Functions
- Selectable Interrupts
- On-Chip Primary and Secondary Address Recognition
- Automatic Handling of Addressing and Handshake Protocol
- Provision for Software Implementation of Additional Features
- 1-5 MHz Clock Range
- 16 Registers (8 Read, 8 Write), 2 for Data Transfer, the Rest for Interface Function Control, Status, etc.
- Directly Interfaces to External Non-Inverting Transceivers for Connection to the GPIB
- Provides Three Addressing Modes, Allowing the Chip to be Addressed Either as a Major or a Minor Talker/Listener with Primary or Secondary Addressing
- DMA Handshake Provision Allows for Bus Transfers without CPU Intervention
- Trigger Output Pin
- On-Chip EOS (End of Sequence) Message Recognition Facilitates Handling of Multi-Byte Transfers

The 8291 GPIB Talker/Listener is a microprocessor-controlled chip designed to interface microprocessors (e.g., 8080, 8085, 8086, 8048) to an IEEE Standard 488 Instrumentation Interface Bus. It implements all of the Standard's Talker/Listener interface functions.

## PIN CONFIGURATION



## BLOCK DIAGRAM



# 8292 GPIB CONTROLLER

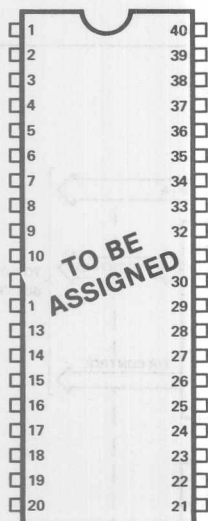
**PRELIMINARY**  
Notice: This is not a final specification. Some parametric limits are subject to change.

## FEATURES:

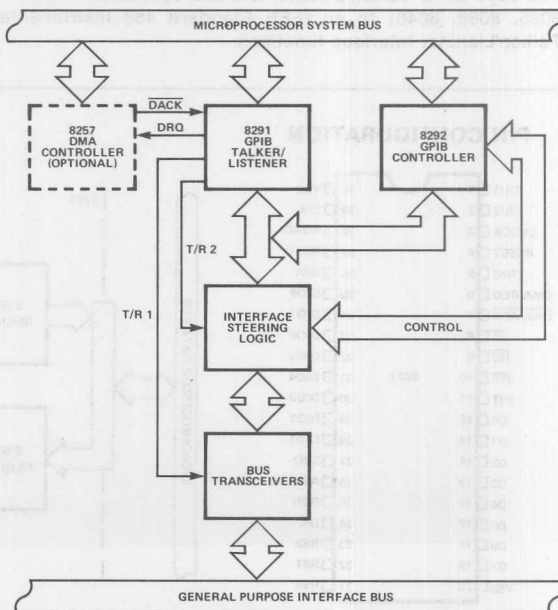
- Complete IEEE Standard 488 Controller Function.
- Interface Clear (IFC) Sending Capability Allows for Seizure of Control and/or Initialization of the Bus.
- Responds to Service Requests (SRQ).
- Sends (REN), Allowing Instruments to Switch to Remote Control.
- Complete Implementation of Transfer Control Protocol.
- Synchronous Control Seizure Prevents the Destruction of any Data Transmission in Progress.
- Connects with the 8291 to Form a Complete IEEE Standard 488 Interface Talker/Listener/Controller.

The 8292 GPIB CONTROLLER is a microprocessor-controlled chip designed to connect with the 8291 GPIB TALKER/LISTENER to implement the full IEEE Standard 488 controller function, including transfer control protocol.

## PIN CONFIGURATION



## 8291, 8292 SYSTEM DIAGRAM





## 8294 DATA ENCRYPTION UNIT

**PRELIMINARY**  
Notice: This is not a final specification. Some parametric limits are subject to change.

- Certified by National Bureau of Standards
- 80-Byte/Sec Data Conversion Rate
- 64-Bit Data Encryption Using 56-Bit Key
- DMA Interface
- 3 Interrupt Outputs to Aid in Loading and Unloading Data
- 7-Bit User Output Port
- Single 5V  $\pm$  10% Power Supply
- Peripheral to MCS-85<sup>TM</sup>, MCS-80<sup>TM</sup> and MCS-48<sup>TM</sup> Processors
- Implements Federal Information Processing Data Encryption Standard
- Encrypt and Decrypt Modes Available

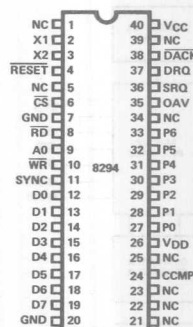
### DESCRIPTION

The Intel® 8294 Data Encryption Unit (DEU) is a microprocessor peripheral device designed to encrypt and decrypt 64-bit blocks of data using the algorithm specified in the Federal Information Processing Data Encryption Standard. The DEU operates on 64-bit text words using a 56-bit user-specified key to produce 64-bit cipher words. The operation is reversible: if the cipher word is operated upon, the original text word is produced. The algorithm itself is permanently contained in the 8294; however, the 56-bit key is user-defined and may be changed at any time.

The 56-bit key and 64-bit message data are transferred to and from the 8294 in 8-bit bytes by way of the system data bus. A DMA interface and three interrupt outputs are available to minimize software overhead associated with data transfer. Also, by using the DMA interface two or more DEUs may be operated in parallel to achieve effective system conversion rates which are virtually any multiple of 120 bytes/second. The 8294 also has a 7-bit TTL compatible output port for user-specified functions.

Because the 8294 implements the NBS encryption algorithm it can be used in a variety of Electronic Funds Transfer applications as well as other electronic banking and data handling applications where data must be encrypted.

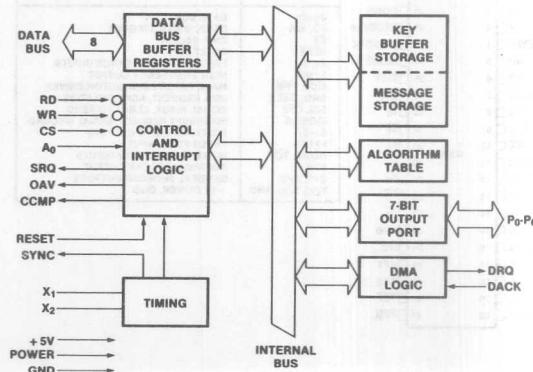
### PIN CONFIGURATION



### PIN NAMES

PIN NAME	FUNCTION
D7-D0	DATA BUS
RD, WR	READ, WRITE STROBES
CS	CHIP SELECT
A0	CONTROL/DATA SELECT
RESET	RESET INPUT
X1, X2	FREQUENCY REFERENCE INPUT
SYNC	HIGH FREQUENCY OUTPUT
DRQ, DACK	DMA REQUEST, DMA ACKNOWLEDGE
SRQ, OAV, CCMP	INTERRUPT REQUEST OUTPUTS
P0-P6	OUTPUT PORT LINES
VCC, VDD, GND	+5V POWER, GND

### BLOCK DIAGRAM



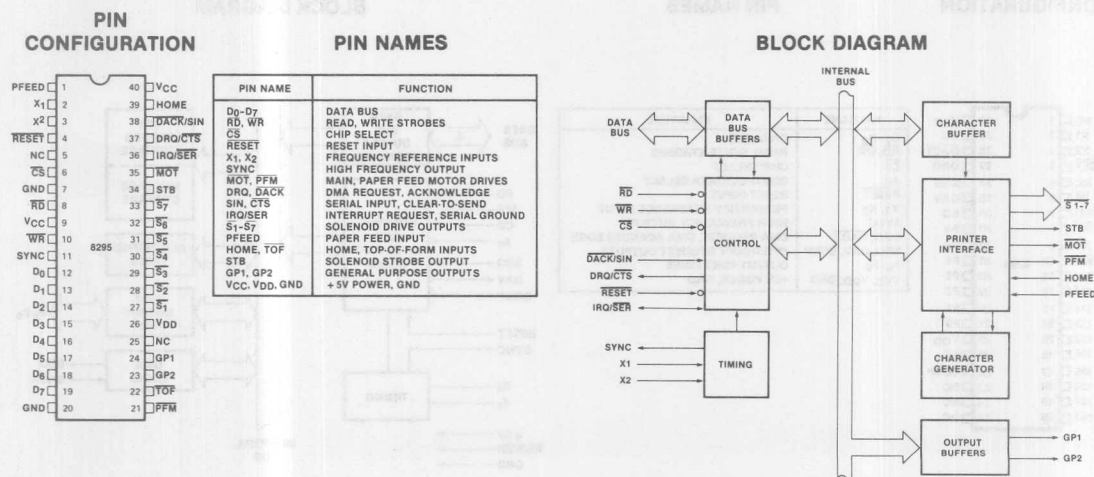
# 8295 DOT MATRIX PRINTER CONTROLLER

**ADVANCE INFORMATION**  
Characteristics are subject to change without notice.

- Interfaces Dot Matrix Printers to MCS-48™, MCS-80™, MCS-85™ Systems
- 40 Character Buffer On Chip
- Serial or Parallel Communication with Host
- DMA Transfer Capability
- Programmable Character Density (10 or 12 Characters/Inch)
- Programmable Print Intensity
- Single or Double Width Printing
- Programmable Multiple Line Feeds
- 3 Tabulations
- 2 General Purpose Outputs

The Intel® 8295 Dot Matrix Printer Controller provides an interface for microprocessors to the LRC 7040 Series dot matrix impact printers. It may also be used as an interface to other similar printers.

The chip may be used in a serial or parallel communication mode with the host processor. Furthermore, it provides internal buffering of up to 40 characters and contains a 7 × 7 matrix character generator accommodating 64 ASCII characters.



## CHAPTER 5

**MCS-85**  
**System Support**  
**Components**  
**Peripherals**  
**Static RAMs**  
**ROMs/EPROMs**

**MCS**  
**85**

**MCS**  
**85**

**MCS**  
**85**





## 2114 1024 X 4 BIT STATIC RAM

	2114-2	2114-3	2114	2114L2	2114L3	2114L
Max. Access Time (ns)	200	300	450	200	300	450
Max. Power Dissipation (mw)	525	525	525	370	370	370

- High Density 18 Pin Package
- Identical Cycle and Access Times
- Single +5V Supply
- No Clock or Timing Strobe Required
- Completely Static Memory
- Directly TTL Compatible: All Inputs and Outputs
- Common Data Input and Output Using Three-State Outputs
- Pin-Out Compatible with 3605 and 3625 Bipolar PROMs

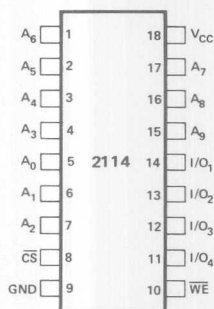
The Intel® 2114 is a 4096-bit static Random Access Memory organized as 1024 words by 4-bits using N-channel Silicon-Gate MOS technology. It uses fully DC stable (static) circuitry throughout — in both the array and the decoding — and therefore requires no clocks or refreshing to operate. Data access is particularly simple since address setup times are not required. The data is read out nondestructively and has the same polarity as the input data. Common input/output pins are provided.

The 2114 is designed for memory applications where high performance, low cost, large bit storage, and simple interfacing are important design objectives. The 2114 is placed in an 18-pin package for the highest possible density.

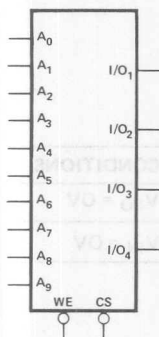
It is directly TTL compatible in all respects: inputs, outputs, and a single +5V supply. A separate Chip Select ( $\overline{CS}$ ) lead allows easy selection of an individual package when outputs are or-tied.

The 2114 is fabricated with Intel's N-channel Silicon-Gate technology — a technology providing excellent protection against contamination permitting the use of low cost plastic packaging.

### PIN CONFIGURATION



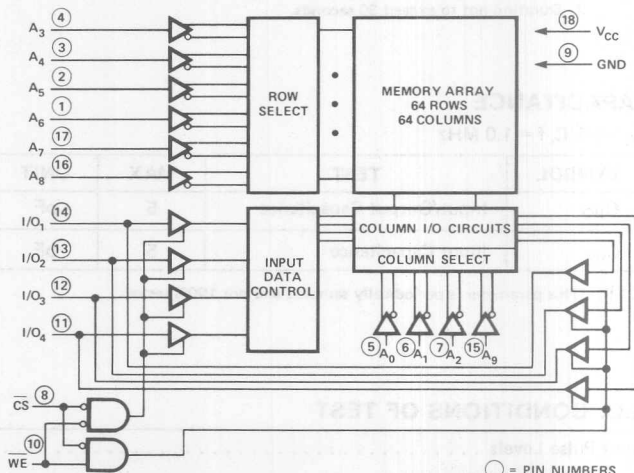
### LOGIC SYMBOL



### PIN NAMES

A <sub>0</sub> -A <sub>9</sub>	ADDRESS INPUTS	V <sub>CC</sub> POWER (+5V)
WE	WRITE ENABLE	GND GROUND
CS	CHIP SELECT	
I/O <sub>1</sub> -I/O <sub>4</sub>	DATA INPUT/OUTPUT	

### BLOCK DIAGRAM



○ = PIN NUMBERS

**ABSOLUTE MAXIMUM RATINGS\***

Temperature Under Bias	-10°C to 80°C
Storage Temperature	-65°C to +150°C
Voltage on Any Pin	
With Respect to Ground	-0.5V to +7V
Power Dissipation	1.0W
D.C. Output Current	5mA

\*COMMENT: Stresses above those listed under "Absolute Maximum Ratings" may cause permanent damage to the device. This is a stress rating only and functional operation of the device at these or any other conditions above those indicated in the operational sections of this specification is not implied. Exposure to absolute maximum rating conditions for extended periods may affect device reliability.

**D.C. AND OPERATING CHARACTERISTICS**

$T_A = 0^\circ\text{C}$  to  $70^\circ\text{C}$ ,  $V_{CC} = 5V \pm 5\%$ , unless otherwise noted.

SYMBOL	PARAMETER	2114-2, 2114-3, 2114			2114L2, 2114L3, 2114L			UNIT	CONDITIONS
		Min.	Typ. <sup>[1]</sup>	Max.	Min.	Typ. <sup>[1]</sup>	Max.		
$I_{LI}$	Input Load Current (All Input Pins)			10			10	$\mu\text{A}$	$V_{IN} = 0$ to $5.25V$
$ I_{LO} $	I/O Leakage Current			10			10	$\mu\text{A}$	$\overline{CS} = 2.4V$ , $V_{I/O} = 0.4V$ to $V_{CC}$
$I_{CC1}$	Power Supply Current		80	95			65	mA	$V_{IN} = 5.25V$ , $I_{I/O} = 0$ mA, $T_A = 25^\circ\text{C}$
$I_{CC2}$	Power Supply Current		100				70	mA	$V_{IN} = 5.25V$ , $I_{I/O} = 0$ mA, $T_A = 0^\circ\text{C}$
$V_{IL}$	Input Low Voltage	-0.5		0.8	-0.5		0.8	V	
$V_{IH}$	Input High Voltage	2.0		6.0	2.0		6.0	V	
$I_{OL}$	Output Low Current	2.1	6.0		2.1	6.0		mA	$V_{OL} = 0.4V$
$I_{OH}$	Output High Current	-1.0	-1.4		-1.0	-1.4		mA	$V_{OH} = 2.4V$
$I_{OS}^{[2]}$	Output Short Circuit Current			40			40	mA	

NOTE: 1. Typical values are for  $T_A = 25^\circ\text{C}$  and  $V_{CC} = 5.0V$ .

2. Duration not to exceed 30 seconds.

**CAPACITANCE**

$T_A = 25^\circ\text{C}$ ,  $f = 1.0$  MHz

SYMBOL	TEST	MAX	UNIT	CONDITIONS
$C_{I/O}$	Input/Output Capacitance	5	pF	$V_{I/O} = 0V$
$C_{IN}$	Input Capacitance	5	pF	$V_{IN} = 0V$

NOTE: This parameter is periodically sampled and not 100% tested.

**A.C. CONDITIONS OF TEST**

Input Pulse Levels	0.8 Volt to 2.4 Volt
Input Rise and Fall Times	10 nsec
Input and Output Timing Levels	1.5 Volts
Output Load	1 TTL Gate and $C_L = 100$ pF

## 2114 FAMILY

**A.C. CHARACTERISTICS**  $T_A = 0^\circ\text{C}$  to  $70^\circ\text{C}$ ,  $V_{CC} = 5V \pm 5\%$ , unless otherwise noted.

### READ CYCLE [1]

SYMBOL	PARAMETER	2114-2, 2114L2		2114-3, 2114L3		2114, 2114L		UNIT
		Min.	Max.	Min.	Max.	Min.	Max.	
$t_{RC}$	Read Cycle Time	200		300		450		ns
$t_A$	Access Time		200		300		450	ns
$t_{CO}$	Chip Selection to Output Valid		70		100		120	ns
$t_{CX}$	Chip Selection to Output Active	20		20		20		ns
$t_{QTD}$	Output 3-state from Deselection		60		80		100	ns
$t_{OHA}$	Output Hold from Address Change	50		50		50		ns

### WRITE CYCLE [2]

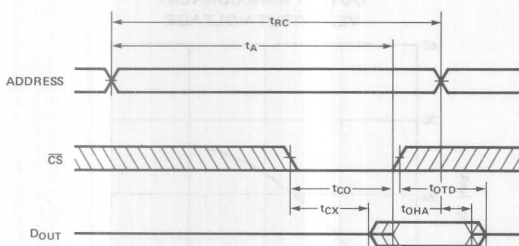
SYMBOL	PARAMETER	2114-2, 2114L2		2114-3, 2114L3		2114, 2114L		UNIT
		Min.	Max.	Min.	Max.	Min.	Max.	
$t_{WC}$	Write Cycle Time	200		300		450		ns
$t_W$	Write Time	120		150		200		ns
$t_{WR}$	Write Release Time	0		0		0		ns
$t_{OTW}$	Output 3-state from Write		60		80		100	ns
$t_{DW}$	Data to Write Time Overlap	120		150		200		ns
$t_{DH}$	Data Hold From Write Time	0		0		0		ns

#### NOTES:

1. A Read occurs during the overlap of a low  $\overline{CS}$  and a high  $\overline{WE}$ .
2. A Write occurs during the overlap of a low  $\overline{CS}$  and a low  $\overline{WE}$ .

## WAVEFORMS

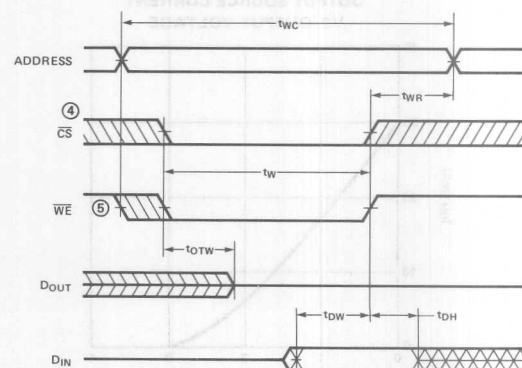
### READ CYCLE ③



#### NOTES:

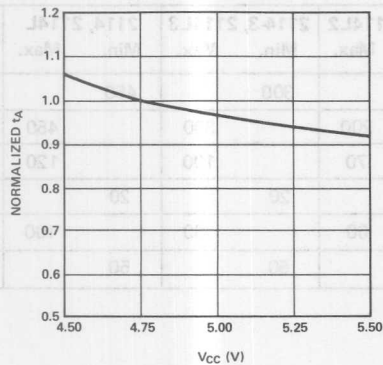
- ③  $\overline{WE}$  is high for a Read Cycle.
- ④ If the  $\overline{CS}$  low transition occurs simultaneously with the  $\overline{WE}$  low transition, the output buffers remain in a high impedance state.
- ⑤  $\overline{WE}$  must be high during all address transitions.

### WRITE CYCLE

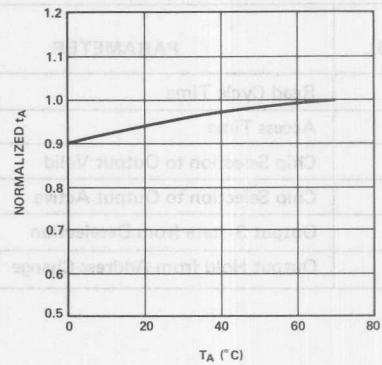


# TYPICAL D.C. AND A.C. CHARACTERISTICS

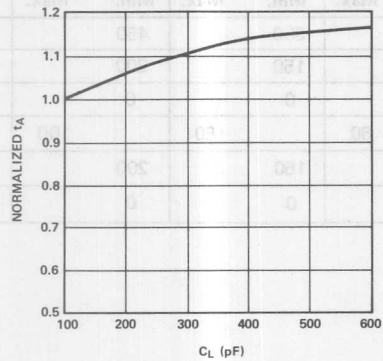
NORMALIZED ACCESS TIME VS. SUPPLY VOLTAGE



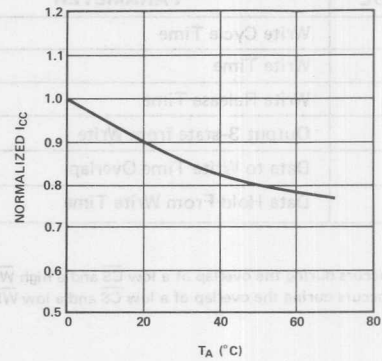
NORMALIZED ACCESS TIME VS. AMBIENT TEMPERATURE



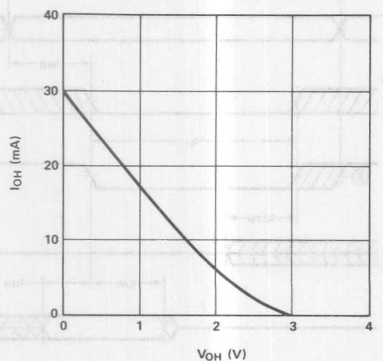
NORMALIZED ACCESS TIME VS. OUTPUT LOAD CAPACITANCE



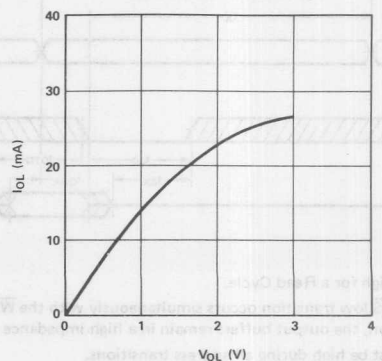
NORMALIZED POWER SUPPLY CURRENT VS. AMBIENT TEMPERATURE



OUTPUT SOURCE CURRENT VS. OUTPUT VOLTAGE



OUTPUT SINK CURRENT VS. OUTPUT VOLTAGE



# M2114 1024 x 4 BIT STATIC RAM

	2114-3	2114	2114L3	2114L
Max. Access Time (ns)	300	450	300	450
Max. Power Dissipation (mw)	575	575	410	410

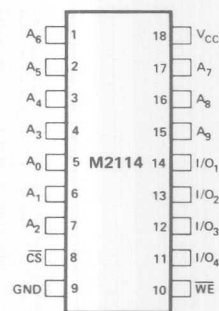
- High Density 18 Pin Package
- Identical Cycle and Access Times
- Single +5V Supply
- No Clock or Timing Strobe Required
- Completely Static Memory
- Directly TTL Compatible: All Inputs and Outputs
- Common Data Input and Output Using Three-State Outputs
- Military Temperature Range -55°C to +125°C

The Intel® M2114 is a 4096-bit static Random Access Memory organized as 1024 words by 4-bits using N-channel Silicon-Gate MOS technology. It uses fully DC stable (static) circuitry throughout — in both the array and the decoding — and therefore requires no clocks or refreshing to operate. Data access is particularly simple since address setup times are not required. The data is read out nondestructively and has the same polarity as the input data. Common input/output pins are provided.

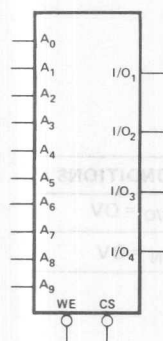
The M2114 is designed for memory applications where high performance, low cost, large bit storage, and simple interfacing are important design objectives. The M2114 is placed in an 18-pin package for the highest possible density.

It is directly TTL compatible in all respects: inputs, outputs, and a single +5V supply. A separate Chip Select ( $\overline{CS}$ ) lead allows easy selection of an individual package when outputs are OR-tied.

## PIN CONFIGURATION



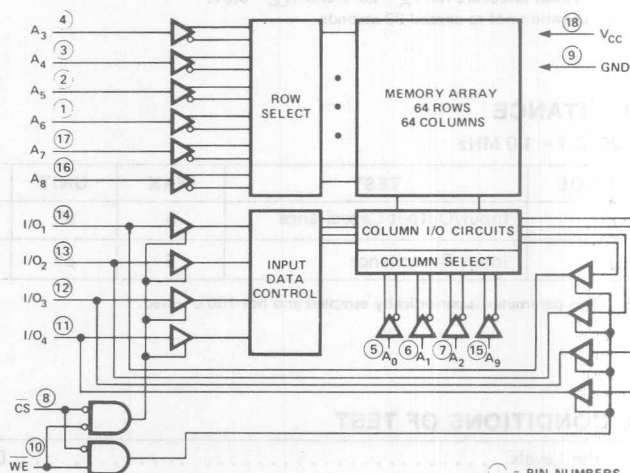
## LOGIC SYMBOL



## PIN NAMES

A <sub>0</sub> -A <sub>9</sub>	ADDRESS INPUTS	V <sub>CC</sub> POWER (+5V)
WE	WRITE ENABLE	GND GROUND
$\overline{CS}$	CHIP SELECT	
I/O <sub>1</sub> -I/O <sub>4</sub>	DATA INPUT/OUTPUT	

## BLOCK DIAGRAM



○ = PIN NUMBERS

## M2114 FAMILY

### ABSOLUTE MAXIMUM RATINGS\*

Temperature Under Bias	-65°C to +150°C
Storage Temperature	-65°C to +150°C
Voltage on Any Pin	
With Respect to Ground	-0.5V to +7V
Power Dissipation	1.0W
D.C. Output Current	5mA

\*COMMENT: Stresses above those listed under "Absolute Maximum Ratings" may cause permanent damage to the device. This is a stress rating only and functional operation of the device at these or any other conditions above those indicated in the operational sections of this specification is not implied. Exposure to absolute maximum rating conditions for extended periods may affect device reliability.

### D.C. AND OPERATING CHARACTERISTICS

T<sub>A</sub> = -55°C to +125°C, V<sub>CC</sub> = 5V ± 10%, unless otherwise noted.

SYMBOL	PARAMETER	M2114, M2114-3			M2114L3, M2114L3			UNIT	CONDITIONS
		Min.	Typ. <sup>[1]</sup>	Max.	Min.	Typ. <sup>[1]</sup>	Max.		
I <sub>LI</sub>	Input Load Current (All Input Pins)			10			10	μA	V <sub>IN</sub> = 0 to 5.5V
I <sub>LO</sub>	I/O Leakage Current			10			10	μA	$\overline{CS}$ = 2.4V, V <sub>I/O</sub> = 0.4V to V <sub>CC</sub>
I <sub>CC1</sub>	Power Supply Current	80		95			65	mA	V <sub>IN</sub> = 5.5V, I <sub>I/O</sub> = 0 mA, T <sub>A</sub> = 25°C
I <sub>CC2</sub>	Power Supply Current			105			75	mA	V <sub>IN</sub> = 5.5V, I <sub>I/O</sub> = 0 mA, T <sub>A</sub> = -55°C
V <sub>IL</sub>	Input Low Voltage	-0.5		0.8	-0.5		0.8	V	
V <sub>IH</sub>	Input High Voltage	2.0		6.0	2.0		6.0	V	
I <sub>OL</sub>	Output Low Current	2.1	6.0		2.1	6.0		mA	V <sub>OL</sub> = 0.4V
I <sub>OH</sub>	Output High Current	-1.0	-1.4		-1.0	-1.4		mA	V <sub>OH</sub> = 2.4V
I <sub>OS</sub> <sup>[2]</sup>	Output Short Circuit Current			40			40	mA	

NOTE: 1. Typical values are for T<sub>A</sub> = 25°C and V<sub>CC</sub> = 5.0V.

2. Duration not to exceed 30 seconds.

### CAPACITANCE

T<sub>A</sub> = 25°C, f = 1.0 MHz

SYMBOL	TEST	MAX	UNIT	CONDITIONS
C <sub>I/O</sub>	Input/Output Capacitance	5	pF	V <sub>I/O</sub> = 0V
C <sub>IN</sub>	Input Capacitance	5	pF	V <sub>IN</sub> = 0V

NOTE: This parameter is periodically sampled and not 100% tested.

### A.C. CONDITIONS OF TEST

Input Pulse Levels	0.8 Volt to 2.0 Volt
Input Rise and Fall Times	10 nsec
Input and Output Timing Levels	1.5 Volts
Output Load	1 TTL Gate and C <sub>L</sub> = 100 pF

# A.C. CHARACTERISTICS

$T_A = -55^\circ\text{C}$  to  $+125^\circ\text{C}$ ,  $V_{CC} = 5\text{V} \pm 10\%$ , unless otherwise noted.

## READ CYCLE [1]

SYMBOL	PARAMETER	M2114, M2114L		M2114-3, M2114L3		UNIT
		Min.	Max.	Min.	Max.	
$t_{RC}$	Read Cycle Time	450		300		ns
$t_A$	Access Time		450		300	ns
$t_{CO}$	Chip Selection to Output Valid		120		100	ns
$t_{CX}$	Chip Selection to Output Active	20		20		ns
$t_{OTD}$	Output 3-state from Deselection		100		80	ns
$t_{OHA}$	Output Hold from Address Change	50		50		ns

## WRITE CYCLE [2]

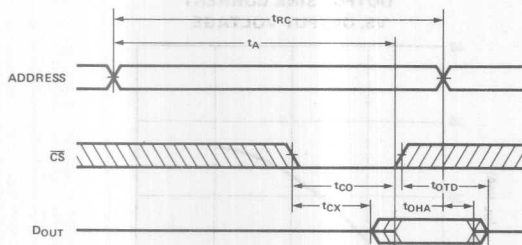
SYMBOL	PARAMETER	M2114, M2114L		M2114-3, M2114L3		UNIT
		Min.	Max.	Min.	Max.	
$t_{WC}$	Write Cycle Time	450		300		ns
$t_W$	Write Time	200		150		ns
$t_{WR}$	Write Release Time	0		0		ns
$t_{OTW}$	Output 3-state from Write		100		80	ns
$t_{DW}$	Data to Write Time Overlap	200		150		ns
$t_{DH}$	Data Hold From Write Time	0		0		ns

### NOTES:

1. A Read occurs during the overlap of a low  $\overline{CS}$  and a high  $\overline{WE}$ .
2. A Write occurs during the overlap of a low  $\overline{CS}$  and a low  $\overline{WE}$ .

## WAVEFORMS

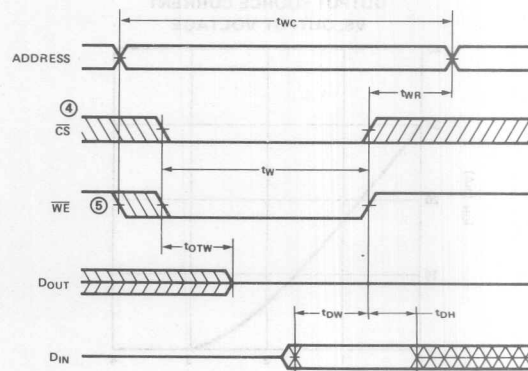
### READ CYCLE ③



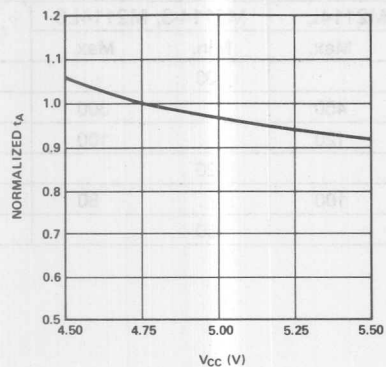
### NOTES:

- ③  $\overline{WE}$  is high for a Read Cycle.
- ④ If the  $\overline{CS}$  low transition occurs simultaneously with the  $\overline{WE}$  low transition, the output buffers remain in a high impedance state.
- ⑤  $\overline{WE}$  must be high during all address transitions.

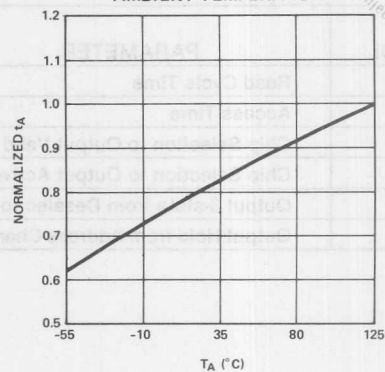
### WRITE CYCLE



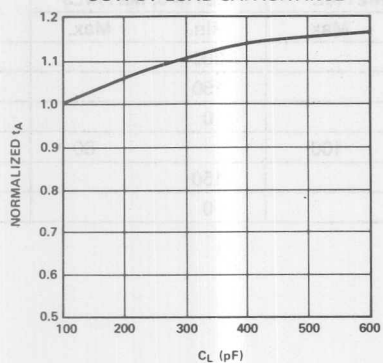
**NORMALIZED ACCESS TIME VS.  
SUPPLY VOLTAGE**



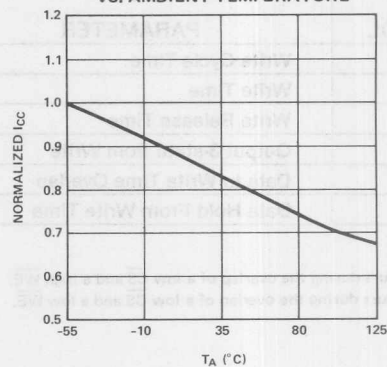
**NORMALIZED ACCESS TIME VS.  
AMBIENT TEMPERATURE**



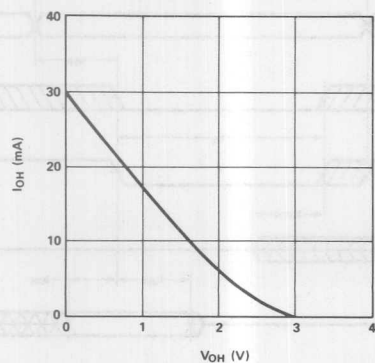
**NORMALIZED ACCESS TIME VS.  
OUTPUT LOAD CAPACITANCE**



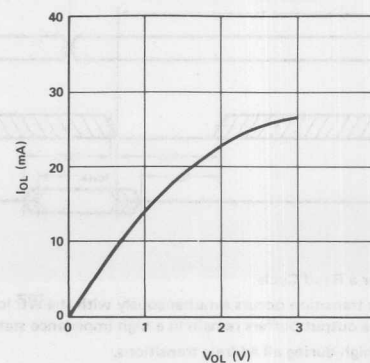
**NORMALIZED POWER SUPPLY CURRENT  
VS. AMBIENT TEMPERATURE**



**OUTPUT SOURCE CURRENT  
VS. OUTPUT VOLTAGE**



**OUTPUT SINK CURRENT  
VS. OUTPUT VOLTAGE**



# 2142

## 1024 X 4 BIT STATIC RAM

	2142-2	2142-3	2142	2142L2	2142L3	2142L
Max. Access Time (ns)	200	300	450	200	300	450
Max. Power Dissipation (mw)	525	525	525	370	370	370

- High Density 20 Pin Package
- Access Time Selections From 200-450ns
- Identical Cycle and Access Times
- Low Operating Power Dissipation  
.1mW/Bit Typical
- Single +5V Supply
- No Clock or Timing Strobe Required
- Completely Static Memory
- Directly TTL Compatible: All Inputs and Outputs
- Common Data Input and Output Using Three-State Outputs

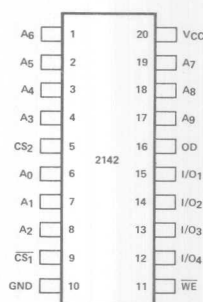
The Intel® 2142 is a 4096-bit static Random Access Memory organized as 1024 words by 4-bits using N-channel Silicon-Gate MOS technology. It uses fully DC stable (static) circuitry throughout — in both the array and the decoding — and therefore requires no clocks or refreshing to operate. Data access is particularly simple since address setup times are not required. The data is read out nondestructively and has the same polarity as the input data. Common input/output pins are provided.

The 2142 is designed for memory applications where high performance, low cost, large bit storage, and simple interfacing are important design objectives. It is directly TTL compatible in all respects: inputs, outputs, and a single +5V supply.

The 2142 is placed in a 20-pin package. Two Chip Selects ( $\overline{CS}_1$  and  $\overline{CS}_2$ ) are provided for easy and flexible selection of individual packages when outputs are OR-tied. An Output Disable is included for direct control of the output buffers.

The 2142 is fabricated with Intel's N-channel Silicon-Gate technology — a technology providing excellent protection against contamination permitting the use of low cost plastic packaging.

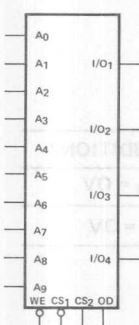
### PIN CONFIGURATION



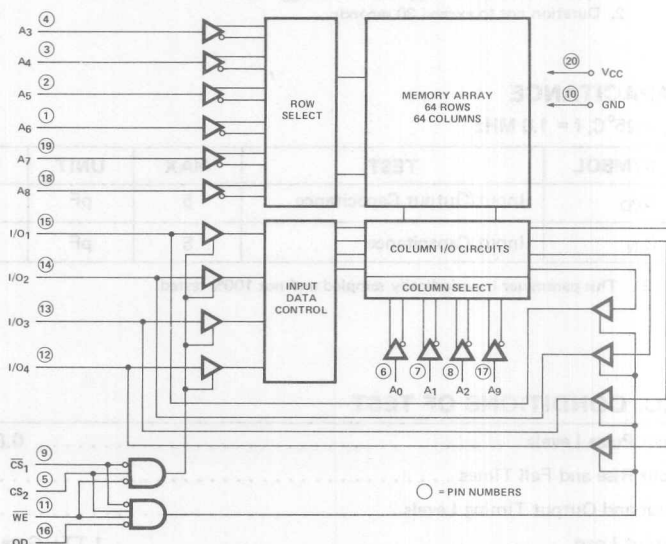
### PIN NAMES

A <sub>0</sub> —A <sub>9</sub>	ADDRESS INPUTS	OD	OUTPUT DISABLE
WE	WRITE ENABLE	V <sub>CC</sub>	POWER (+5V)
$\overline{CS}_1, \overline{CS}_2$	CHIP SELECT	GND	GROUND
I/O <sub>1</sub> —I/O <sub>4</sub>	DATA INPUT/OUTPUT		

### LOGIC SYMBOL



### BLOCK DIAGRAM



**ABSOLUTE MAXIMUM RATINGS\***

Temperature Under Bias	-10°C to 80°C
Storage Temperature	-65°C to +150°C
Voltage on Any Pin	
With Respect to Ground	-0.5V to +7V
Power Dissipation	1.0W
D.C. Output Current	10mA

\*COMMENT: Stresses above those listed under "Absolute Maximum Ratings" may cause permanent damage to the device. This is a stress rating only and functional operation of the device at these or any other conditions above those indicated in the operational sections of this specification is not implied. Exposure to absolute maximum rating conditions for extended periods may affect device reliability.

**D.C. AND OPERATING CHARACTERISTICS**

$T_A = 0^\circ\text{C}$  to  $70^\circ\text{C}$ ,  $V_{CC} = 5\text{V} \pm 5\%$ , unless otherwise noted.

SYMBOL	PARAMETER	2142-2, 2142-3, 2142			2142L2, 2142L3, 2142L			UNIT	CONDITIONS
		Min.	Typ. <sup>[1]</sup>	Max.	Min.	Typ. <sup>[1]</sup>	Max.		
$I_{LI}$	Input Load Current (All Input Pins)			10			10	$\mu\text{A}$	$V_{IN} = 0$ to $5.25\text{V}$
$ I_{LO} $	I/O Leakage Current			10			10	$\mu\text{A}$	$\overline{CS} = 2.4\text{V}$ , $V_{I/O} = 0.4\text{V}$ to $V_{CC}$
$I_{CC1}$	Power Supply Current		80	95			65	mA	$V_{IN} = 5.25\text{V}$ , $I_{I/O} = 0$ mA, $T_A = 25^\circ\text{C}$
$I_{CC2}$	Power Supply Current			100			70	mA	$V_{IN} = 5.25\text{V}$ , $I_{I/O} = 0$ mA, $T_A = 0^\circ\text{C}$
$V_{IL}$	Input Low Voltage	-0.5		0.8	-0.5		0.8	V	
$V_{IH}$	Input High Voltage	2.0		6.0	2.0		6.0	V	
$I_{OL}$	Output Low Current	2.1	6.0		2.1	6.0		mA	$V_{OL} = 0.4\text{V}$
$I_{OH}$	Output High Current	-1.0	-1.4		-1.0	-1.4		mA	$V_{OH} = 2.4\text{V}$
$I_{OS}^{[2]}$	Output Short Circuit Current			40			40	mA	$V_{I/O} = \text{GND to } V_{CC}$

NOTE: 1. Typical values are for  $T_A = 25^\circ\text{C}$  and  $V_{CC} = 5.0\text{V}$ .

2. Duration not to exceed 30 seconds.

**CAPACITANCE**

$T_A = 25^\circ\text{C}$ ,  $f = 1.0\text{ MHz}$

SYMBOL	TEST	MAX	UNIT	CONDITIONS
$C_{I/O}$	Input/Output Capacitance	5	pF	$V_{I/O} = 0\text{V}$
$C_{IN}$	Input Capacitance	5	pF	$V_{IN} = 0\text{V}$

NOTE: This parameter is periodically sampled and not 100% tested.

**A.C. CONDITIONS OF TEST**

Input Pulse Levels	0.8 Volt to 2.4 Volt
Input Rise and Fall Times	10 nsec
Input and Output Timing Levels	1.5 Volts
Output Load	1 TTL Gate and $C_L = 100\text{ pF}$

## 2142 FAMILY

### A.C. CHARACTERISTICS $T_A = 0^\circ\text{C}$ to $70^\circ\text{C}$ , $V_{CC} = 5V \pm 5\%$ , unless otherwise noted.

#### READ CYCLE [1]

SYMBOL	PARAMETER	2142-2, 2142L2		2142-3, 2142L3		2142, 2142L		UNIT
		Min.	Max.	Min.	Max.	Min.	Max.	
$t_{RC}$	Read Cycle Time	200		300		450		ns
$t_A$	Access Time		200		300		450	ns
$t_{OD}$	Output Enable to Output Valid		70		100		120	ns
$t_{ODX}$	Output Enable to Output Active	20		20		20		ns
$t_{CO}$	Chip Selection to Output Valid		70		100		120	ns
$t_{CX}$	Chip Selection to Output Active	20		20		20		ns
$t_{OTD}$	Output 3-state from Disable		60		80		100	ns
$t_{OHA}$	Output Hold from Address Change	50		50		50		ns

#### WRITE CYCLE [2]

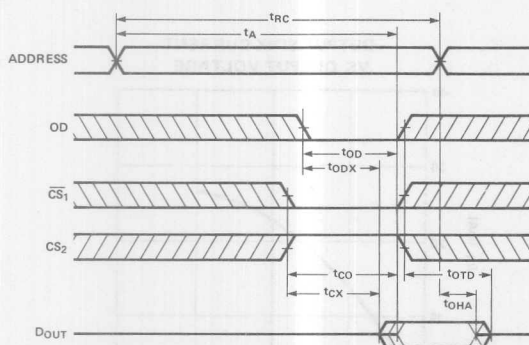
SYMBOL	PARAMETER	2142-2, 2142L2		2142-3, 2142L3		2142, 2142L		UNIT
		Min.	Max.	Min.	Max.	Min.	Max.	
$t_{WC}$	Write Cycle Time	200		300		450		ns
$t_W$	Write Time	120		150		200		ns
$t_{WR}$	Write Release Time	0		0		0		ns
$t_{OTD}$	Output 3-state from Disable		60		80		100	ns
$t_{DW}$	Data to Write Time Overlap	120		150		200		ns
$t_{DH}$	Data Hold From Write Time	0		0		0		ns

#### NOTES:

1. A Read occurs during the overlap of a low  $\overline{CS}$  and a high  $\overline{WE}$ .
2. A Write occurs during the overlap of a low  $\overline{CS}$  and a low  $\overline{WE}$ .

### WAVEFORMS

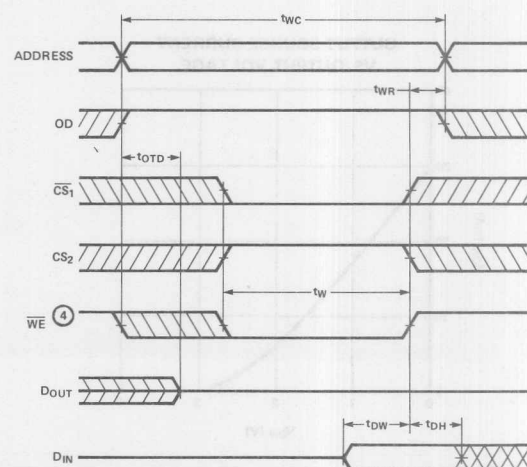
#### READ CYCLE ③



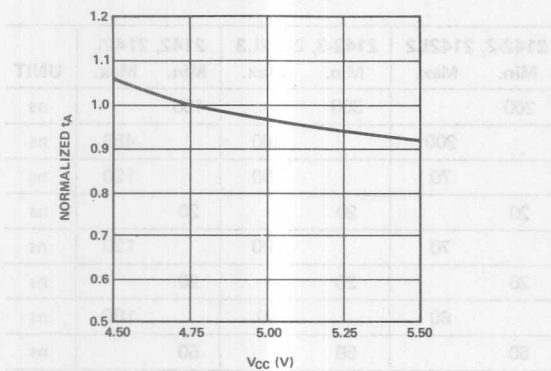
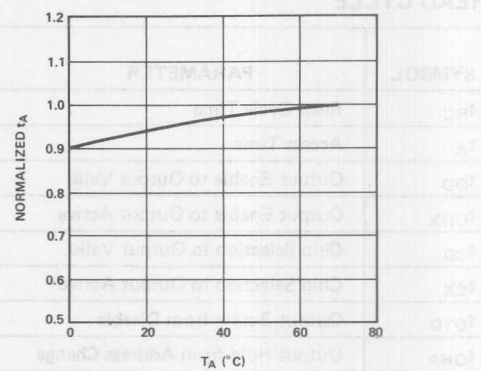
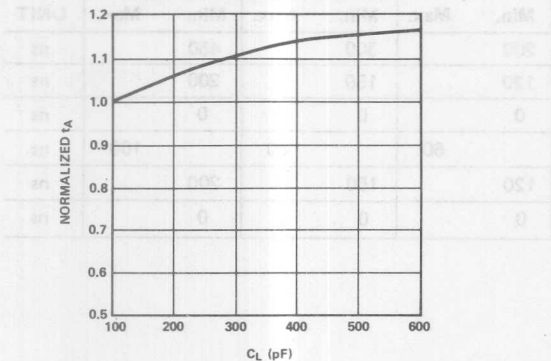
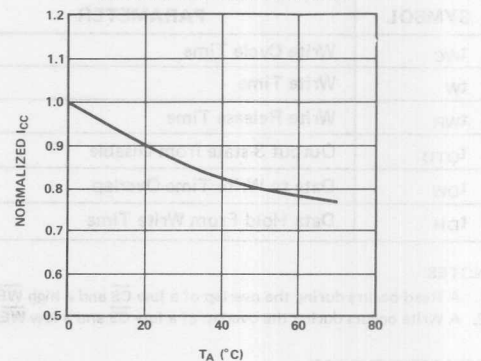
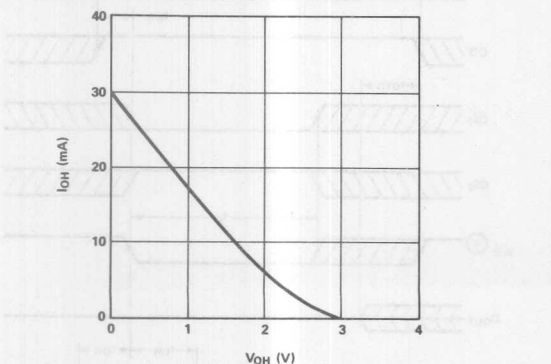
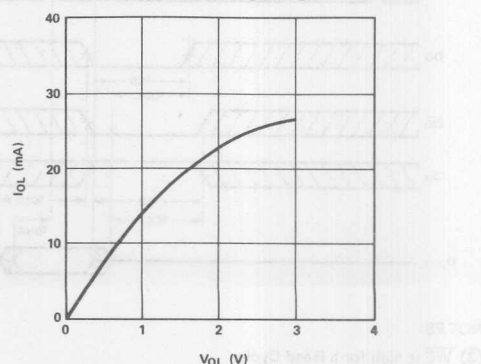
#### NOTES:

- ③  $\overline{WE}$  is high for a Read Cycle.
- ④  $\overline{WE}$  must be high during all address transitions.

#### WRITE CYCLE



## TYPICAL D.C. AND A.C. CHARACTERISTICS

NORMALIZED ACCESS TIME VS.  
SUPPLY VOLTAGENORMALIZED ACCESS TIME VS.  
AMBIENT TEMPERATURENORMALIZED ACCESS TIME VS.  
OUTPUT LOAD CAPACITANCENORMALIZED POWER SUPPLY CURRENT  
VS. AMBIENT TEMPERATUREOUTPUT SOURCE CURRENT  
VS. OUTPUT VOLTAGEOUTPUT SINK CURRENT  
VS. OUTPUT VOLTAGE

# CHAPTER 5

**MCS-85**  
**System Support**  
**Components**  
**Peripherals**  
**Static RAMs**  
**ROMs/EPROMs**

**MCS**  
**85**

**MCS**  
**85**

**MCS**  
**85**



# 2332 32K (4K x 8) ROM

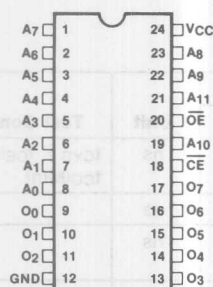
- Single +5V  $\pm$  10% Power Supply
- Pin Compatible to Intel® 2716 and 2732 EPROMs
- 300ns Max. Access Time
- Low Power Dissipation:  
40mA Max. Average Current  
15mA Max. Standby Current
- Edge Enabled With Static Array
- Inputs and Outputs TTL Compatible
- Three-State Output for Direct Bus Interface
- Output Enable for MCS-85™ and MCS-86™ Compatibility

The Intel® 2332 is a single +5V supply, 32,768-bit N-channel MOS read only memory organized as 4096 words by 8-bits. It has static memory cells and clocked peripheral circuitry, giving a fast device access time with low active power dissipation. The 2332 features an automatic standby power mode. When deselected by  $\overline{CE}$ , the active power dissipation is reduced from 40mA to 15mA, a 60% reduction.

The 2332 is ideal for microprocessor systems, especially those with common input and output bus structures. The separate output control,  $\overline{OE}$ , eliminates bus contention. The 300ns access time, three-state outputs, address latches, and TTL input/output levels further simplify system design.

A cost effective system development program may be implemented by using the pin compatible Intel® 2732, 32K UV EPROM for prototyping and the 2332 ROM for volume production. The 2732 is fully compatible to the 2332 in all respects.

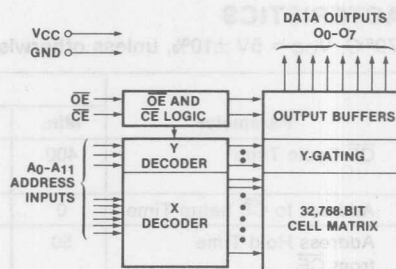
## PIN CONFIGURATION



## PIN NAMES

A0-A10	ADDRESSES
CE	CHIP ENABLE
OE	OUTPUT ENABLE
O0-O7	OUTPUTS

## BLOCK DIAGRAM



**ABSOLUTE MAXIMUM RATINGS\***

Ambient Temperature Under Bias .... -10°C to 80°C  
 Storage Temperature ..... -65°C to +150°C  
 Voltage On Any Pin with Respect  
 to Ground ..... -0.5V to +7V

\*COMMENT: Stresses above those listed under "Absolute Maximum Ratings" may cause permanent damage to the device. This is a stress rating only and functional operation of the device at these or any other conditions above those indicated in the operational sections of this specification is not implied. Exposure to absolute maximum rating conditions for extended periods may affect device reliability.

**D.C. AND OPERATING CHARACTERISTICS**

T<sub>A</sub> = 0°C to +70°C, V<sub>CC</sub> = 5V ±10%, unless otherwise specified.

Symbol	Parameter	Limits			Unit	Test Conditions
		Min.	Typ. <sup>(1)</sup>	Max.		
I <sub>LI</sub>	Input Load Current (All Input Pins)			10	μA	V <sub>IN</sub> = 0 to 5.5V
I <sub>LOH</sub>	Output Leakage Current			10	μA	Chip Deselected, V <sub>OUT</sub> = 5.5V
I <sub>LOL</sub>	Output Leakage Current			-20	μA	Chip Deselected, V <sub>OUT</sub> = 0.4V
I <sub>CC1</sub>	V <sub>CC</sub> Standby Current			15	mA	$\overline{CE}$ = 2V
I <sub>CC2</sub>	V <sub>CC</sub> Average Current			40	mA	t <sub>CYC</sub> = 400ns
V <sub>IL</sub>	Input "Low" Voltage	-0.5		0.8	V	
V <sub>IH</sub>	Input "High" Voltage	2.0		V <sub>CC</sub> +1.0V	V	
V <sub>OL</sub>	Output "Low" Voltage			0.4	V	I <sub>OL</sub> = 3.2mA
V <sub>OH</sub>	Output "High" Voltage	2.4			V	I <sub>OH</sub> = -400μA

NOTE: 1. Typical for T<sub>A</sub> = 25°C and nominal supply voltage.

**A.C. CHARACTERISTICS**

T<sub>A</sub> = 0°C to +70°C, V<sub>CC</sub> = 5V ±10%, unless otherwise specified.

Symbol	Parameter	Limits			Unit	Test Conditions
		Min.	Typ. <sup>(1)</sup>	Max.		
t <sub>CYC</sub>	$\overline{CE}$ Cycle Time	400			ns	t <sub>CYC</sub> = t <sub>CE</sub> (Max) + t <sub>CC</sub> (Min)
t <sub>AS</sub>	Address to $\overline{CE}$ Setup Time	0			ns	
t <sub>AH</sub>	Address Hold Time from $\overline{CE}$	50			ns	
t <sub>CE</sub>	$\overline{CE}$ to Output Delay			300	ns	
t <sub>OE</sub>	$\overline{OE}$ to Output Delay			120	ns	
t <sub>ACC</sub>	Address to Output Delay			300	ns	t <sub>AS</sub> =0ns, t <sub>OE</sub> =120ns
t <sub>CC</sub>	$\overline{CE}$ Off Time	100			ns	
t <sub>DF</sub>	$\overline{OE}$ to Data Float	0		70	ns	

## CONDITIONS OF TEST FOR A.C. CHARACTERISTICS

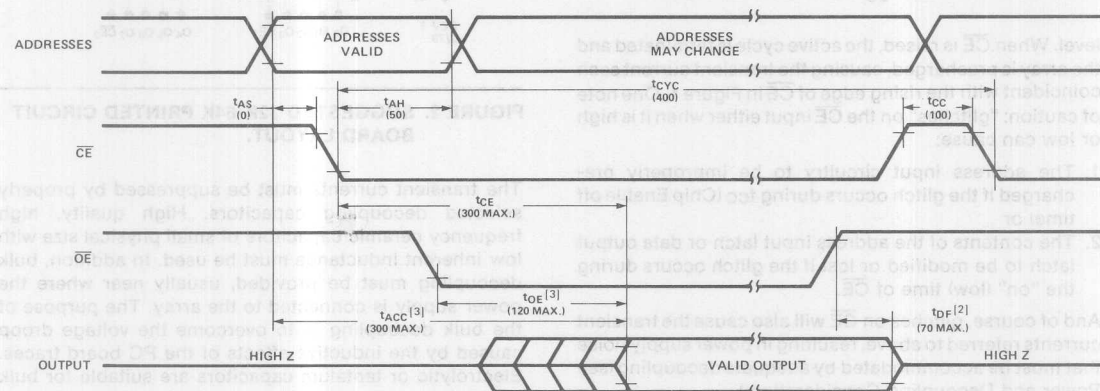
Output Load . . . . . 1 TTL Gate and  $C_L = 100$  pF  
 Input Pulse Levels . . . . . 0.8 to 2.4V  
 Input Pulse Rise and Fall Times (10% to 90%) . . . . . 20 ns  
 Timing Measurement Reference Level  
   Input . . . . . 1V and 2.2V  
   Output . . . . . 0.8V and 2.0V

## CAPACITANCE<sup>[1]</sup> $T_A = 25^\circ\text{C}$ , $f = 1$ MHz

SYMBOL	TEST	LIMITS	
		TYP.	MAX.
$C_{IN}$	All Pins Except Pin Under Test Tied to AC Ground	5 pF	10 pF
$C_{OUT}$	All Pins Except Pin Under Test Tied to AC Ground	10 pF	15 pF

NOTE: 1. This parameter is periodically sampled and is not 100% tested.

## A.C. WAVEFORMS<sup>[1]</sup>



### NOTES:

1. ALL TIMES SHOWN IN PARENTHESES ARE MINIMUM TIMES AND ARE nsec UNLESS OTHERWISE SPECIFIED.
2.  $t_{DF}$  IS SPECIFIED FROM OE OR CE, WHICHEVER OCCURS FIRST.
3. OE MAY BE DELAYED UP TO 180ns AFTER THE FALLING EDGE OF CE WITHOUT IMPACT ON  $t_{ACC}$ .

## DEVICE DESCRIPTION

Unlike other Intel® MOS ROMs, the 2332 is a clocked or Edge Enabled device. Clocked devices have a substantial speed power advantage over completely static devices of similar density. Both the active as well as the standby power is reduced in a clocked device as a result of the dynamic periphery. The dynamic periphery is also inherently faster than the circuitry used in a comparable static device.

The basic difference between completely static and clocked devices is the fact that access time ( $t_{ACC}$ ) does not equal cycle time, ( $t_{CYC}$ ). The reason for this is that the clock, in this case CE, has a minimum "off" or HIGH time, which is the period during which the non-static periphery (address input buffers, latches etc.) is being precharged. This "off" period ( $t_{CC}$  in the timing diagram) may be extended indefinitely but violation of the minimum time will result in inadequate precharge causing the addresses to be improperly latched, usually resulting in invalid data out. It is also permissible to maintain CE "on", i.e., low, indefinitely.

An access cycle proceeds as follows. Addresses must be

presented prior to Chip Enable ( $\overline{CE}$ ) going low. The falling edge of Chip Enable activates the address input buffers and latches the addresses in preparation for the address decoders and sense amplifiers to perform their function. This activity is responsible for both the transient current and the increase in current from standby to active, shown coincident with the falling edge of  $\overline{CE}$  in Figure 1. The addresses must be held stable for a minimum of  $t_{AH}$  after the falling edge of  $\overline{CE}$ . After  $t_{AH}$  they have been latched and the input buffers have been disabled, so the addresses can change without affecting the data that will be sensed.

The internal cycle then proceeds, resulting in the data being latched in the output buffer. The falling edge of OE (Output Enable) enables the data to be presented, by way of the output buffer, to the output pin after  $t_{OE}$ , the output buffer delay time. The output will remain stable as long as OE remains low; the state of the output is maintained by active internal transistors so that eventual "droop" will not be experienced. The output is caused to go to a high impedance state by raising either CE or OE to a HIGH level; in either case the start of the output float delay,  $t_{DF}$ , applies to the first signal (OE or CE) that is raised to a high

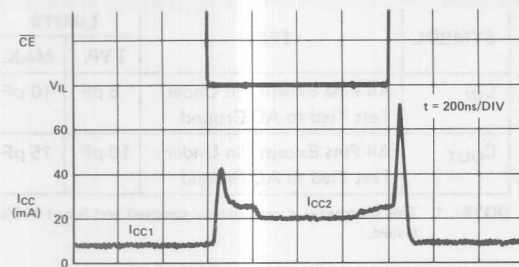


FIGURE 1. TYPICAL  $I_{CC}$  CURRENT VS. TIME.

level. When  $\overline{CE}$  is raised, the active cycle is terminated and the array is precharged, causing the transient current seen coincident with the rising edge of  $\overline{CE}$  in Figure 1. One note of caution: "glitches" on the  $\overline{CE}$  input either when it is high or low can cause:

1. The address input circuitry to be improperly precharged if the glitch occurs during  $t_{CC}$  (Chip Enable off time) or
2. The contents of the address input latch or data output latch to be modified or lost if the glitch occurs during the "on" (low) time of  $\overline{CE}$ .

And of course, glitches on  $\overline{CE}$  will also cause the transient currents referred to above, resulting in power supply noise that must be accommodated by adequate decoupling (see Power and Decoupling Consideration.)

### POWER SUPPLY DECOUPLING/ DISTRIBUTION

The  $I_{CC}$  waveform for the 2332 is shown in Figure 1, and specified in the DC Operating Characteristic Table. The supply current,  $I_{CC}$ , has three segments that are of concern to the system designer: standby level, active level, and the transient current peaks that are produced on the rising and falling edges of Chip Enable. The D.C. segments of the current ( $I_{CC1}$  and  $I_{CC2}$ ) are specified as a maximum average value and should be used to determine power supply ampacity. Note that cycle time should also be taken into consideration in this determination.

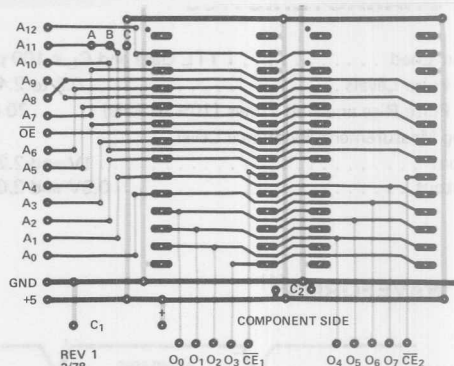


FIGURE 2. SUGGESTED 32K/64K PRINTED CIRCUIT BOARD LAYOUT.

The transient currents must be suppressed by properly selected decoupling capacitors. High quality, high frequency ceramic capacitors of small physical size with low inherent inductance must be used. In addition, bulk decoupling must be provided, usually near where the power supply is connected to the array. The purpose of the bulk decoupling is to overcome the voltage droop caused by the inductive effects of the PC board traces. Electrolytic or tantalum capacitors are suitable for bulk decoupling.

The following capacitance values and locations are recommended for the 2332:

1. A  $0.1\mu F$  ceramic capacitor between  $V_{CC}$  and GND at every other device.
2. A  $4.7\mu F$  electrolytic capacitor between  $V_{CC}$  and GND for each eight devices.

Intel recommends a power supply distribution system such that the power supply and ground traces on the PC board are gridded both vertically and horizontally at each memory device; this technique minimizes the power distribution system impedance and enhances the effect of the decoupling capacitors.

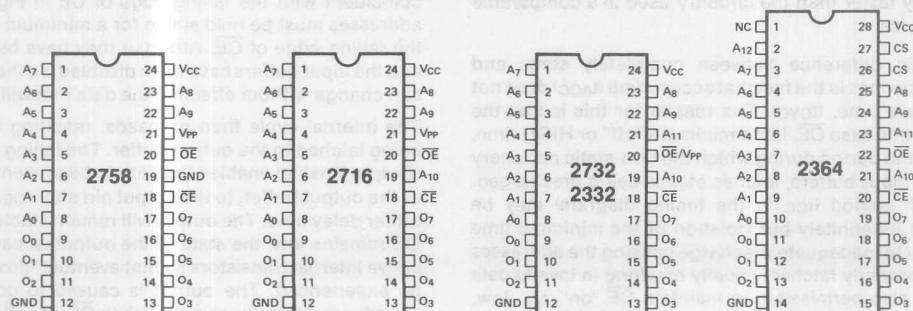


FIGURE 3. THE COMPATIBLE EPROM/ROM FAMILY.

# 2364 64K (8K x 8) ROM

- Single +5V  $\pm$  10% Power Supply
- Pin Compatible to Intel® 2716/2732 EPROMs and 2332 ROM
- 300ns Max. Access Time
- Low Power Dissipation:  
40mA Max. Average Current  
15mA Max. Standby Current

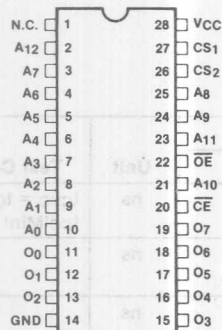
- Edge Enabled With Static Array
- Inputs and Outputs TTL Compatible
- Three-State Output for Direct Bus Interface
- Output Enable for MCS-85™ and MCS-86™ Compatibility

The Intel® 2364 is a single +5V supply, 65,536-bit N-channel MOS read only memory organized as 8192 words by 8-bits. It has static memory cells and clocked peripheral circuitry, giving a fast device access time with low active power dissipation. The 2364 features an automatic standby power mode. When deselected by  $\overline{CE}$ , the active power dissipation is reduced from 40mA to 15mA, a 60% reduction.

The 2364 is ideal for microprocessor systems, especially those with common input and output bus structures. The separate output control,  $\overline{OE}$ , eliminates bus contention. The 300ns access time, three-state outputs, address latches, and TTL input/output levels further simplify system design.

A cost-effective system development program may be implemented by using the Intel® 2716/2732 UV EPROMs for prototyping and the 2364 ROM for production. The lower 24 pins of the 2364 are the same as the EPROM to facilitate board designs in making the transition from EPROM to ROM.

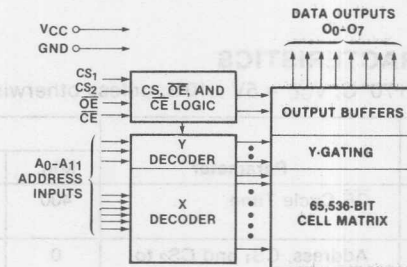
## PIN CONFIGURATION



## PIN NAMES

A0-A12	ADDRESSES
OE	OUTPUT ENABLE
CE	CHIP ENABLE
CS	CHIP SELECT
N.C.	NO CONNECTION

## BLOCK DIAGRAM



**ABSOLUTE MAXIMUM RATINGS\***

Ambient Temperature Under Bias ..... -10°C to 80°C  
 Storage Temperature ..... -65°C to +150°C  
 Voltage On Any Pin with Respect  
 to Ground ..... -0.5V to +7V

\*COMMENT: Stresses above those listed under "Absolute Maximum Ratings" may cause permanent damage to the device. This is a stress rating only and functional operation of the device at these or any other conditions above those indicated in the operational sections of this specification is not implied. Exposure to absolute maximum rating conditions for extended periods may affect device reliability.

**D.C. AND OPERATING CHARACTERISTICS**

T<sub>A</sub> = 0°C to +70°C, V<sub>CC</sub> = 5V ±10%, unless otherwise specified.

Symbol	Parameter	Limits			Unit	Test Conditions
		Min.	Typ. <sup>(1)</sup>	Max.		
I <sub>LI</sub>	Input Load Current (All Input Pins)			10	μA	V <sub>IN</sub> = 0 to 5.5V
I <sub>LOH</sub>	Output Leakage Current			10	μA	Chip Deselected, V <sub>OUT</sub> = 5.5V
I <sub>LOL</sub>	Output Leakage Current			-20	μA	Chip Deselected, V <sub>OUT</sub> = 0.4V
I <sub>CC1</sub>	V <sub>CC</sub> Standby Current			15	mA	$\overline{CE}$ = 2V
I <sub>CC2</sub>	V <sub>CC</sub> Average Current			40	mA	t <sub>CYC</sub> = 400ns
V <sub>IL</sub>	Input "Low" Voltage	-0.5		0.8	V	
V <sub>IH</sub>	Input "High" Voltage	2.0		V <sub>CC</sub> +1.0V	V	
V <sub>OL</sub>	Output "Low" Voltage			0.4	V	I <sub>OL</sub> = 3.2mA
V <sub>OH</sub>	Output "High" Voltage	2.4			V	I <sub>OH</sub> = -400μA

NOTE: 1. Typical for T<sub>A</sub> = 25°C and nominal supply voltage.

**A.C. CHARACTERISTICS**

T<sub>A</sub> = 0°C to +70°C, V<sub>CC</sub> = 5V ±10%, unless otherwise specified.

Symbol	Parameter	Limits			Unit	Test Conditions
		Min.	Typ. <sup>(1)</sup>	Max.		
t <sub>CYC</sub>	$\overline{CE}$ Cycle Time	400			ns	t <sub>CYC</sub> = t <sub>CE</sub> (Max) + t <sub>CC</sub> (Min)
t <sub>AS</sub>	Address, CS <sub>1</sub> and CS <sub>2</sub> to $\overline{CE}$ Setup Time	0			ns	
t <sub>AH</sub>	Address, CS <sub>1</sub> and CS <sub>2</sub> Hold Time from $\overline{CE}$	50			ns	
t <sub>CE</sub>	$\overline{CE}$ to Output Delay			300	ns	
t <sub>OE</sub>	$\overline{OE}$ to Output Delay			120	ns	
t <sub>ACC</sub>	Address to Output Delay			300	ns	t <sub>AS</sub> =0ns, t <sub>OE</sub> =120ns
t <sub>CC</sub>	$\overline{CE}$ Off Time	100			ns	
t <sub>DF</sub>	$\overline{OE}$ to Data Float	0		70	ns	

## CONDITIONS OF TEST FOR A.C. CHARACTERISTICS

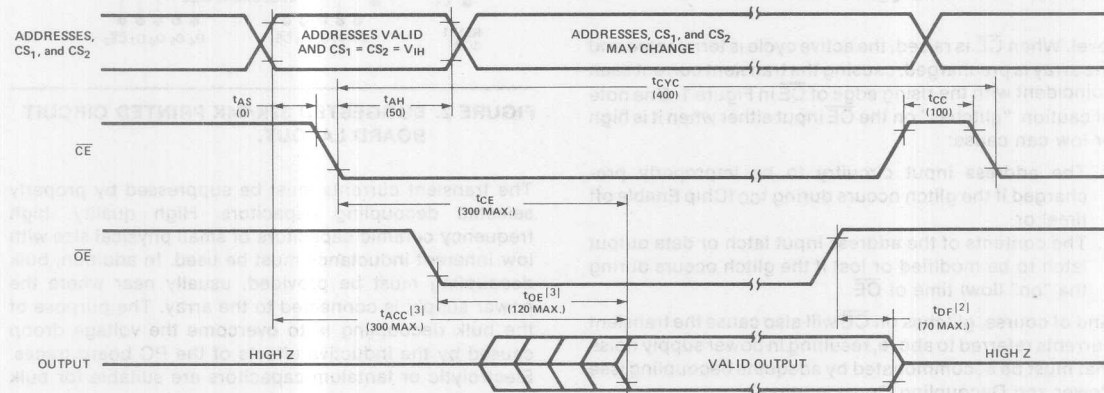
Output Load . . . . . 1 TTL Gate and  $C_L = 100$  pF  
 Input Pulse Levels . . . . . 0.8 to 2.4V  
 Input Pulse Rise and Fall Times (10% to 90%) . . . . . 20 ns  
 Timing Measurement Reference Level  
   Input . . . . . 1V and 2.2V  
   Output . . . . . 0.8V and 2.0V

## CAPACITANCE<sup>[1]</sup> $T_A = 25^\circ\text{C}$ , $f = 1$ MHz

SYMBOL	TEST	LIMITS	
		TYP.	MAX.
$C_{IN}$	All Pins Except Pin Under Test Tied to AC Ground	5 pF	10 pF
$C_{OUT}$	All Pins Except Pin Under Test Tied to AC Ground	10 pF	15 pF

NOTE: 1. This parameter is periodically sampled and is not 100% tested.

## A.C. WAVEFORMS<sup>[1]</sup>



### NOTES:

- ALL TIMES SHOWN IN PARENTHESES ARE MINIMUM TIMES AND ARE nsec UNLESS OTHERWISE SPECIFIED.
- $t_{DF}$  IS SPECIFIED FROM  $\overline{OE}$  OR  $\overline{CE}$ , WHICHEVER OCCURS FIRST.
- $t_{ACC}$  MAY BE DELAYED UP TO 180ns AFTER THE FALLING EDGE OF  $\overline{CE}$  WITHOUT IMPACT ON  $t_{ACC}$ .

## DEVICE DESCRIPTION

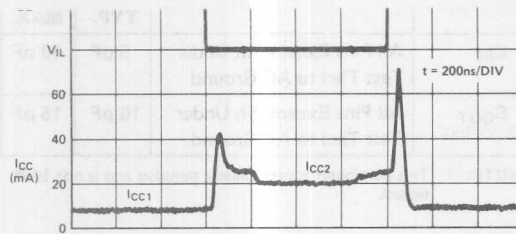
Unlike other Intel® MOS ROMs, the 2364 is a clocked or Edge Enabled device. Clocked devices have a substantial speed power advantage over completely static devices of similar density. Both the active as well as the standby power is reduced in a clocked device as a result of the dynamic periphery. The dynamic periphery is also inherently faster than the circuitry used in a comparable static device.

The basic difference between completely static and clocked devices is the fact that access time ( $t_{ACC}$ ) does not equal cycle time, ( $t_{CYC}$ ). The reason for this is that the clock, in this case  $\overline{CE}$ , has a minimum "off" or HIGH time, which is the period during which the non-static periphery (address input buffers, latches etc.) is being precharged. This "off" period ( $t_{CC}$  in the timing diagram) may be extended indefinitely but violation of the minimum time will result in inadequate precharge causing the addresses to be improperly latched, usually resulting in invalid data out. It is also permissible to maintain  $\overline{CE}$  "on", i.e., LOW, indefinitely.

An access cycle proceeds as follows. Addresses and Chip Selects must be presented prior to Chip Enable ( $\overline{CE}$ ) going

low. The falling edge of Chip Enable activates the address input buffers and latches the addresses and chip selects in preparation for the decoders and sense amplifiers to perform their function. This activity is responsible for both the transient current and the increase in current from standby to active, shown coincident with the falling edge of  $\overline{CE}$  in Figure 1. The addresses and chip selects must be held stable for a minimum of  $t_{AH}$  after the falling edge of  $\overline{CE}$ . After  $t_{AH}$  they have been latched and the input buffers have been disabled, so the addresses and chip selects can change without affecting the data that will be sensed.

The internal cycle then proceeds, resulting in the data being latched in the output buffer. The falling edge of  $\overline{OE}$  (Output Enable) enables the data to be presented, by way of the output buffer, to the output pin after  $t_{OE}$ , the output buffer delay time. The output will remain stable as long as  $\overline{OE}$  remains low; the state of the output is maintained by active internal transistors so that eventual "droop" will not be experienced. The output is caused to go to a high impedance state by raising either  $\overline{CE}$  or  $\overline{OE}$  to a HIGH level; in either case the start of the output float delay,  $t_{DF}$ , applies to the first signal ( $\overline{OE}$  or  $\overline{CE}$ ) that is raised to a high

FIGURE 1. TYPICAL  $I_{CC}$  CURRENT VS. TIME.

level. When  $\overline{CE}$  is raised, the active cycle is terminated and the array is precharged, causing the transient current seen coincident with the rising edge of  $\overline{CE}$  in Figure 1. One note of caution: "glitches" on the  $\overline{CE}$  input either when it is high or low can cause:

1. The address input circuitry to be improperly precharged if the glitch occurs during  $t_{CC}$  (Chip Enable off time) or
2. The contents of the address input latch or data output latch to be modified or lost if the glitch occurs during the "on" (low) time of  $\overline{CE}$ .

And of course, glitches on  $\overline{CE}$  will also cause the transient currents referred to above, resulting in power supply noise that must be accommodated by adequate decoupling (see Power and Decoupling Consideration.)

### POWER SUPPLY DECOUPLING/ DISTRIBUTION

The  $I_{CC}$  waveform for the 2364 is shown in Figure 1, and specified in the DC Operating Characteristic Table. The supply current,  $I_{CC}$ , has three segments that are of concern to the system designer: standby level, active level, and the transient current peaks that are produced on the rising and falling edges of Chip Enable. The D.C. segments of the current ( $I_{CC1}$  and  $I_{CC2}$ ) are specified as a maximum average value and should be used to determine power supply ampacity. Note that cycle time should also be taken into consideration in this determination.

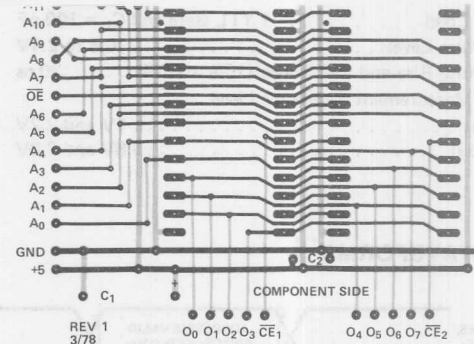


FIGURE 2. SUGGESTED 32K/64K PRINTED CIRCUIT BOARD LAYOUT.

The transient currents must be suppressed by properly selected decoupling capacitors. High quality, high frequency ceramic capacitors of small physical size with low inherent inductance must be used. In addition, bulk decoupling must be provided, usually near where the power supply is connected to the array. The purpose of the bulk decoupling is to overcome the voltage droop caused by the inductive effects of the PC board traces. Electrolytic or tantalum capacitors are suitable for bulk decoupling.

The following capacitance values and locations are recommended for the 2364:

1. A  $0.1\mu F$  ceramic capacitor between  $V_{CC}$  and GND at every other device.
2. A  $4.7\mu F$  electrolytic capacitor between  $V_{CC}$  and GND for each eight devices.

Intel recommends a power supply distribution system such that the power supply and ground traces on the PC board are gridded both vertically and horizontally at each memory device; this technique minimizes the power distribution system impedance and enhances the effect of the decoupling capacitors.

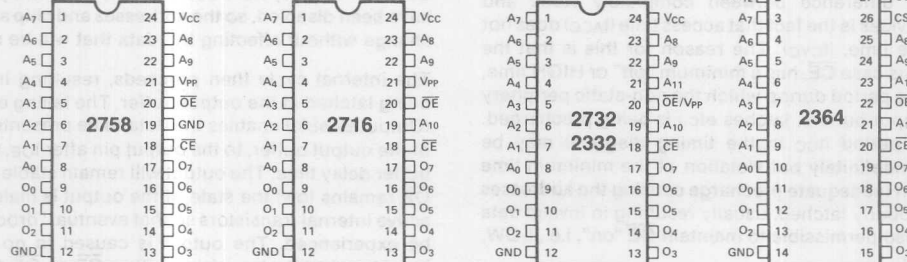


FIGURE 3. THE COMPATIBLE EPROM/ROM FAMILY.

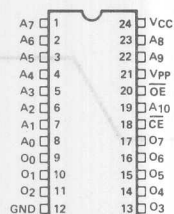
**2616\*****16K (2K × 8) FACTORY PROGRAMMABLE PROM**

- **Single +5V Power Supply**
- **Low Power Dissipation**
  - 525 mW Max. Active Power**
  - 132 mW Max. Standby Power**
- **Pin Compatible to Intel® 2716 EPROM and 2316E ROM**
- **Fast Access Time — 450 ns Max.**
- **Inputs and Outputs TTL Compatible**
- **Completely Static**

The Intel® 2616 is a 16,384-bit, one-time factory-programmable MOS PROM organized as 2048 words by 8 bits. The 2616 operates from a single +5V power supply, has a static standby mode, and is TTL input/output compatible. It is specified over the 0°C to 70°C operating temperature with 5% power supply variation.

A cost-effective system development program may be implemented quickly into production by using the Intel® 2716 EPROM for pattern experimentation, the 2616 for fast first incremental 2316E ROM delivery, and the 2316E for volume production. The 2616 is fully compatible to the 2716 in all respects. The fast factory 2616 code pattern turnaround time gives rapid transition from EPROM to ROM for production.

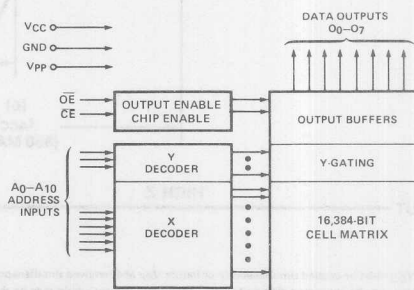
The 2616 has a static standby mode which reduces the power dissipation without increasing access time. The maximum active power dissipation is 525 mW, while the maximum standby power dissipation is only 132 mW — a 75% saving.

**PIN CONFIGURATION\*****PIN NAMES**

A <sub>0</sub> –A <sub>9</sub>	ADDRESSES
CE/PGM	CHIP ENABLE/PROGRAM
OE	OUTPUT ENABLE
O <sub>0</sub> –O <sub>7</sub>	OUTPUTS

**MODE SELECTION**

PINS MODE	$\overline{\text{CE}}$ (18)	$\overline{\text{OE}}$ (20)	V <sub>pp</sub> (21)	V <sub>CC</sub> (24)	OUTPUTS (9-11, 13-17)
Read	V <sub>IL</sub>	V <sub>IL</sub>	+5	+5	D <sub>OUT</sub>
Standby	V <sub>IH</sub>	Don't Care	+5	+5	High Z

**BLOCK DIAGRAM**

\*Pin 18 and pin 20 have been named to conform with the entire family of 16K, 32K, and 64K EPROMs and ROMs.

## A.C. Characteristics

 $T_A = 0^\circ\text{C to } 70^\circ\text{C}$ ,  $V_{CC}^{[1]} = +5\text{V} \pm 5\%$ ,  $V_{PP}^{[2]} = V_{CC} \pm 0.6\text{V}$ 

Symbol	Parameter	Limits			Unit	Test Conditions
		Min.	Typ. <sup>[4]</sup>	Max.		
$t_{ACC}$	Address to Output Delay		250	450	ns	$\overline{CE} = \overline{OE} = V_{IL}$
$t_{CE}$	$\overline{CE}$ to Output Delay		280	450	ns	$\overline{OE} = V_{IL}$
$t_{OE}$	Output Enable to Output Delay			120	ns	$\overline{CE} = V_{IL}$
$t_{DF}$	Output Enable High to Output Float	0		100	ns	$\overline{CE} = V_{IL}$
$t_{OH}$	Address to Output Hold	0			ns	$\overline{CE} = \overline{OE} = V_{IL}$

**Capacitance<sup>[4]</sup>**  $T_A = 25^\circ\text{C}$ ,  $f = 1\text{ MHz}$ 

Symbol	Parameter	Typ.	Max.	Unit	Conditions
$C_{IN}$	Input Capacitance	4	6	pF	$V_{IN} = 0\text{V}$
$C_{OUT}$	Output Capacitance	8	12	pF	$V_{OUT} = 0\text{V}$

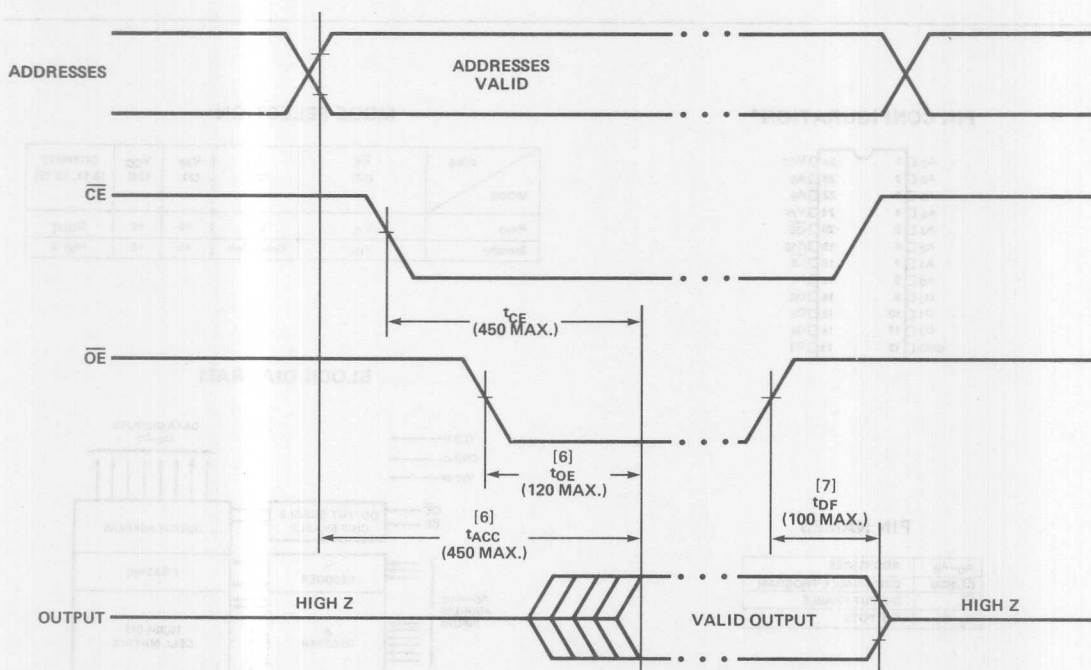
## A.C. Test Conditions:

Output Load: 1 TTL gate and  $C_L = 100\text{ pF}$ Input Rise and Fall Times:  $\leq 20\text{ ns}$ Input Pulse Levels:  $-0.8\text{V to } 2.2\text{V}$ 

Timing Measurement Reference Level:

Inputs 1V and 2V

Outputs 0.8V and 2V

A.C. Waveforms<sup>[5]</sup>

- NOTES:
- $V_{CC}$  must be applied simultaneously or before  $V_{PP}$  and removed simultaneously of after  $V_{PP}$ .
  - $V_{PP}$  may be connected directly to  $V_{CC}$ . The supply current would then be the sum of  $I_{CC}$  and  $I_{PP1}$ .
  - Typical values are for  $T_A = 25^\circ\text{C}$  and nominal supply voltages.
  - This parameter is only sampled and is not 100% tested.
  - All times shown in parentheses are minimum times and are nsec unless otherwise specified.
  - $\overline{OE}$  may be delayed up to 330ns after the falling edge of  $\overline{CE}$  without impact on  $t_{ACC}$ .
  - $t_{DF}$  is specified from  $\overline{OE}$  or  $\overline{CE}$ , whichever occurs first.

### Absolute Maximum Ratings\*

Temperature Under Bias . . . . .  $-10^{\circ}\text{C}$  to  $+80^{\circ}\text{C}$   
 Storage Temperature . . . . .  $-65^{\circ}\text{C}$  to  $+125^{\circ}\text{C}$   
 All Input or Output Voltages with  
 Respect to Ground . . . . .  $+6\text{V}$  to  $-0.3\text{V}$

\*COMMENT: Stresses above those listed under "Absolute Maximum Ratings" may cause permanent damage to the device. This is a stress rating only and functional operation of the device at these or any other conditions above those indicated in the operational sections of this specification is not implied. Exposure to absolute maximum rating conditions for extended periods may affect device reliability.

## READ OPERATION

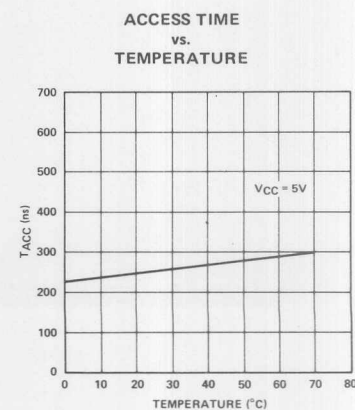
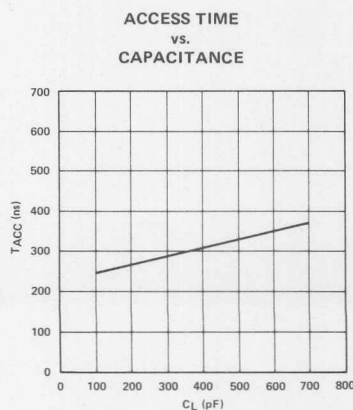
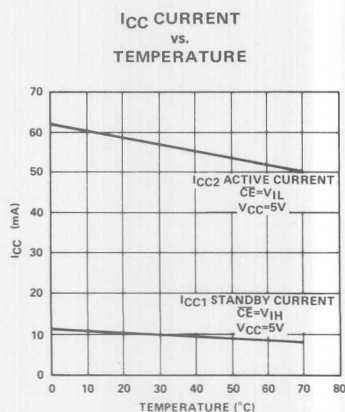
### D.C. and Operating Characteristics

$T_A = 0^{\circ}\text{C}$  to  $70^{\circ}\text{C}$ ,  $V_{CC}^{[1,2]} = +5\text{V} \pm 5\%$ ,  $V_{PP}^{[2]} = V_{CC} \pm 0.6\text{V}$

Symbol	Parameter	Limits			Unit	Conditions
		Min.	Typ. [3]	Max.		
$I_{LI}$	Input Load Current			10	$\mu\text{A}$	$V_{IN} = 5.25\text{V}$
$I_{LO}$	Output Leakage Current			10	$\mu\text{A}$	$V_{OUT} = 5.25\text{V}$
$I_{PP1}^{[2]}$	$V_{PP}$ Current			5	mA	$V_{PP} = 5.85\text{V}$
$I_{CC1}^{[2]}$	$V_{CC}$ Current (Standby)		10	25	mA	$\overline{CE} = V_{IH}$ , $\overline{OE} = V_{IL}$
$I_{CC2}^{[2]}$	$V_{CC}$ Current (Active)		57	100	mA	$\overline{OE} = \overline{CE} = V_{IL}$
$V_{IL}$	Input Low Voltage	-0.1		0.8	V	
$V_{IH}$	Input High Voltage	2.0		$V_{CC}+1$	V	
$V_{OL}$	Output Low Voltage			0.45	V	$I_{OL} = 2.1\text{ mA}$
$V_{OH}$	Output High Voltage	2.4			V	$I_{OH} = -400\text{ }\mu\text{A}$

- NOTES: 1.  $V_{CC}$  must be applied simultaneously or before  $V_{PP}$  and removed simultaneously or after  $V_{PP}$ .  
 2.  $V_{PP}$  may be connected directly to  $V_{CC}$ . The supply current would then be the sum of  $I_{CC}$  and  $I_{PP1}$ .  
 3. Typical values are for  $T_A = 25^{\circ}\text{C}$  and nominal supply voltages.

### Typical Characteristics



## DEVICE OPERATION

The modes of operation of the 2616 are listed in Table I. It should be noted that all inputs are at TTL levels. Only a single +5V power supply is required since  $V_{PP}$  may be connected to  $V_{CC}$ .

TABLE I. MODE SELECTION

MODE	PINS CE (18)	OE (20)	V <sub>PP</sub> (21)	V <sub>CC</sub> (24)	OUTPUTS (9-11, 13-17)
Read	V <sub>IL</sub>	V <sub>IL</sub>	+5	+5	D <sub>OUT</sub>
Standby	V <sub>IH</sub>	Don't Care	+5	+5	High Z

## READ MODE

The 2616 has two control functions, both of which must be logically satisfied in order to obtain data at the outputs. Chip Enable ( $\overline{CE}$ ) is the power control and should be used for device selection. Output Enable ( $\overline{OE}$ ) is the output control and should be used to gate data to the output

pins, independent of device selection. Assuming that addresses are stable, address access time ( $t_{ACC}$ ) is equal to the delay from  $\overline{CE}$  to output ( $t_{CE}$ ). Data is available at the outputs 120 ns ( $t_{OE}$ ) after the falling edge of  $\overline{OE}$ , assuming that  $\overline{CE}$  has been low and addresses have been stable for at least  $t_{ACC} - t_{OE}$ .

## STANDBY MODE

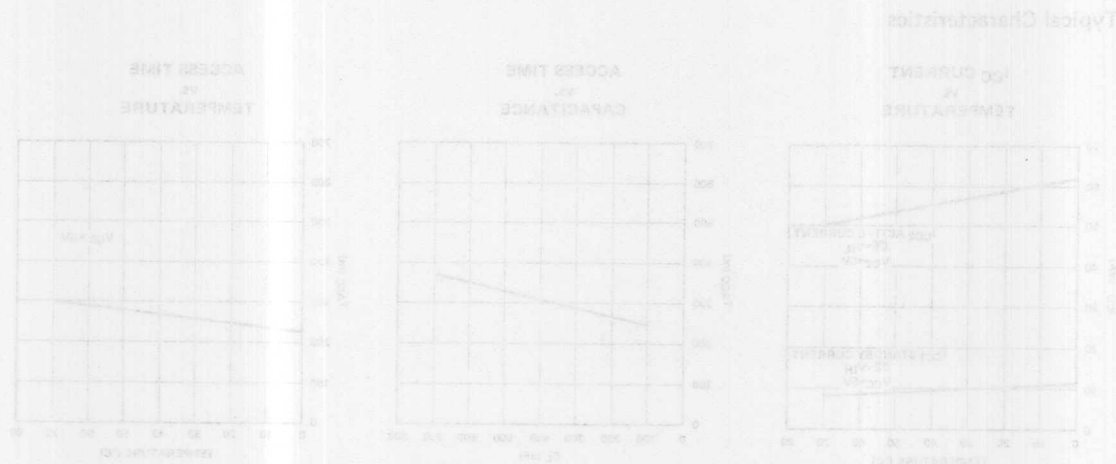
The 2616 has a standby mode which reduces the active power dissipation by 75%, from 525 mW to 132 mW. The 2616 is placed in the standby mode by applying a TTL high signal to the  $\overline{CE}$  input. When in standby mode, the outputs are in a high impedance state, independent of the  $\overline{OE}$  input.

## OUTPUT DESELECTION

The outputs of two or more 2616s may be OR-tied together on the same data bus. Only one 2616 should have its output selected ( $\overline{OE}$  low) to prevent data bus contention between 2616s in this configuration. The outputs of the other 2616s should be deselected by raising the  $\overline{OE}$  input to a TTL high level.

V <sub>OH</sub>	Output High Voltage	2.4	V	I <sub>OH</sub> = -100 $\mu$ A
V <sub>OL</sub>	Output Low Voltage	0.45	V	I <sub>OL</sub> = 5.1 mA
V <sub>IH</sub>	Input High Voltage	2.0	V	V <sub>CC</sub> = 5V
V <sub>IL</sub>	Input Low Voltage	0.8	V	V <sub>CC</sub> = 5V
I <sub>CC</sub> ST	V <sub>CC</sub> Current (Active)	10	mA	V <sub>CE</sub> = V <sub>IL</sub> , V <sub>OE</sub> = V <sub>IL</sub>
I <sub>CC</sub> ST	V <sub>CC</sub> Current (Standby)	2	mA	V <sub>CE</sub> = V <sub>IL</sub> , V <sub>OE</sub> = V <sub>IH</sub>
I <sub>CC</sub> ST	V <sub>PP</sub> Current	2	mA	V <sub>PP</sub> = 5.25V
I <sub>LE</sub>	Output Leakage Current	10	nA	V <sub>CE</sub> = V <sub>OL</sub>
t <sub>CE</sub>	Delay from $\overline{CE}$ to output	120	ns	V <sub>CE</sub> = V <sub>IL</sub> , V <sub>OE</sub> = V <sub>IL</sub>

NOTES: 1. V<sub>CC</sub> must be applied simultaneously or before V<sub>PP</sub> and removed simultaneously or after V<sub>PP</sub>.  
2. V<sub>PP</sub> may be connected directly to V<sub>CC</sub>. The supply current must then be the sum of I<sub>CC</sub> and I<sub>PP</sub>.  
3. Typical values are for T<sub>A</sub> = 25°C and nominal supply voltages.



## 2716\*

## 16K (2K x 8) UV ERASABLE PROM

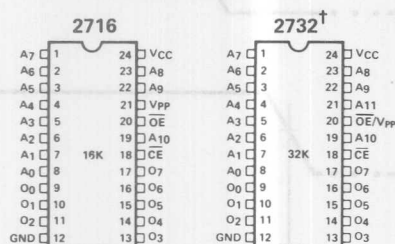
- **Fast Access Time**
  - 350 ns Max. 2716-1
  - 390 ns Max. 2716-2
  - 450 ns Max. 2716
- **Single +5V Power Supply**
- **Low Power Dissipation**
  - 525 mW Max. Active Power
  - 132 mW Max. Standby Power
- **Pin Compatible to Intel® 5V ROMs (2316E, 2332, and 2364) and 2732 EPROM**
- **Simple Programming Requirements**  
Single Location Programming  
Programs with One 50 ms Pulse
- **Inputs and Outputs TTL Compatible during Read and Program**
- **Completely Static**

The Intel® 2716 is a 16,384-bit ultraviolet erasable and electrically programmable read-only memory (EPROM). The 2716 operates from a single 5-volt power supply, has a static standby mode, and features fast single address location programming. It makes designing with EPROMs faster, easier and more economical. For production quantities, the 2716 user can convert rapidly to Intel's pin-for-pin compatible 16K ROM (the 2316E) or the new 32K and 64K ROMs (the 2332 and 2364 respectively).

The 2716, with its single 5-volt supply and with an access time up to 350 ns, is ideal for use with the newer high performance +5V microprocessors such as Intel's 8085 and 8086. The 2716 is also the first EPROM with a static standby mode which reduces the power dissipation without increasing access time. The maximum active power dissipation is 525 mW while the maximum standby power dissipation is only 132 mW, a 75% savings.

The 2716 has the simplest and fastest method yet devised for programming EPROMs — single pulse TTL level programming. No need for high voltage pulsing because all programming controls are handled by TTL signals. Now, it is possible to program on-board, in the system, in the field. Program any location at any time — either individually, sequentially or at random, with the 2716's single address location programming. Total programming time for all 16,384 bits is only 100 seconds.

### PIN CONFIGURATION\*



†Refer to 2732 data sheet for specifications

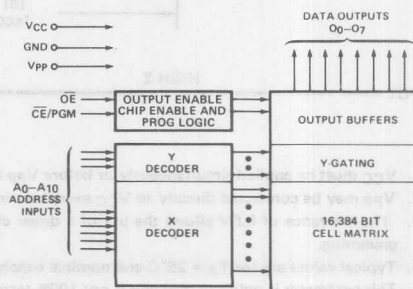
## PIN NAMES

A <sub>0</sub> –A <sub>9</sub>	ADDRESSES
CE/PGM	CHIP ENABLE/PROGRAM
$\overline{OE}$	OUTPUT ENABLE
Q <sub>0</sub> –Q <sub>7</sub>	OUTPUTS

## MODE SELECTION

MODE \ PINS	$\overline{CE}/PGM$ (18)	$\overline{OE}$ (20)	$V_{PP}$ (21)	$V_{CC}$ (24)	OUTPUTS (9-11, 13-17)
Read	$V_{IL}$	$V_{IL}$	+5	+5	$D_{OUT}$
Standby	$V_{IH}$	Don't Care	+5	+5	High Z
Program	Pulsed $V_{IL}$ to $V_{IH}$	$V_{IH}$	+25	+5	$D_{IN}$
Program Verify	$V_{IL}$	$V_{IL}$	+25	+5	$D_{OUT}$
Program Inhibit	$V_{II}$	$V_{IH}$	+25	+5	High Z

### BLOCK DIAGRAM



\*Pin 18 and pin 20 have been renamed to conform with the entire family of 16K, 32K, and 64K EPROMs and ROMs. The die, fabrication process, and specifications remain the same and are totally unaffected by this change.

## A.C. Characteristics

Symbol	Parameter	2716 Limits			2716-1 Limits			2716-2 Limits			Unit	Test Conditions
		Min	Typ <sup>[4]</sup>	Max	Min	Typ <sup>[4]</sup>	Max	Min	Typ <sup>[4]</sup>	Max		
t <sub>ACC</sub>	Address to Output Delay			450			350			390	ns	$\overline{CE} = \overline{OE} = V_{IL}$
t <sub>CE</sub>	$\overline{CE}$ to Output Delay			450			350			390	ns	$\overline{OE} = V_{IL}$
t <sub>OE</sub>	Output Enable to Output Delay			120			120			120	ns	$\overline{CE} = V_{IL}$
t <sub>DF</sub>	Output Enable High to Output Float	0		100	0		100	0		100	ns	$\overline{CE} = V_{IL}$
t <sub>OH</sub>	Address to Output Hold	0			0			0			ns	$\overline{CE} = \overline{OE} = V_{IL}$

Capacitance<sup>[5]</sup>  $T_A = 25^\circ\text{C}$ ,  $f = 1\text{ MHz}$ 

Symbol	Parameter	Typ.	Max.	Unit	Conditions
C <sub>IN</sub>	Input Capacitance	4	6	pF	V <sub>IN</sub> = 0V
C <sub>OUT</sub>	Output Capacitance	8	12	pF	V <sub>OUT</sub> = 0V

## A.C. Test Conditions:

Output Load: 1 TTL gate and C<sub>L</sub> = 100 pFInput Rise and Fall Times:  $\leq 20\text{ ns}$ 

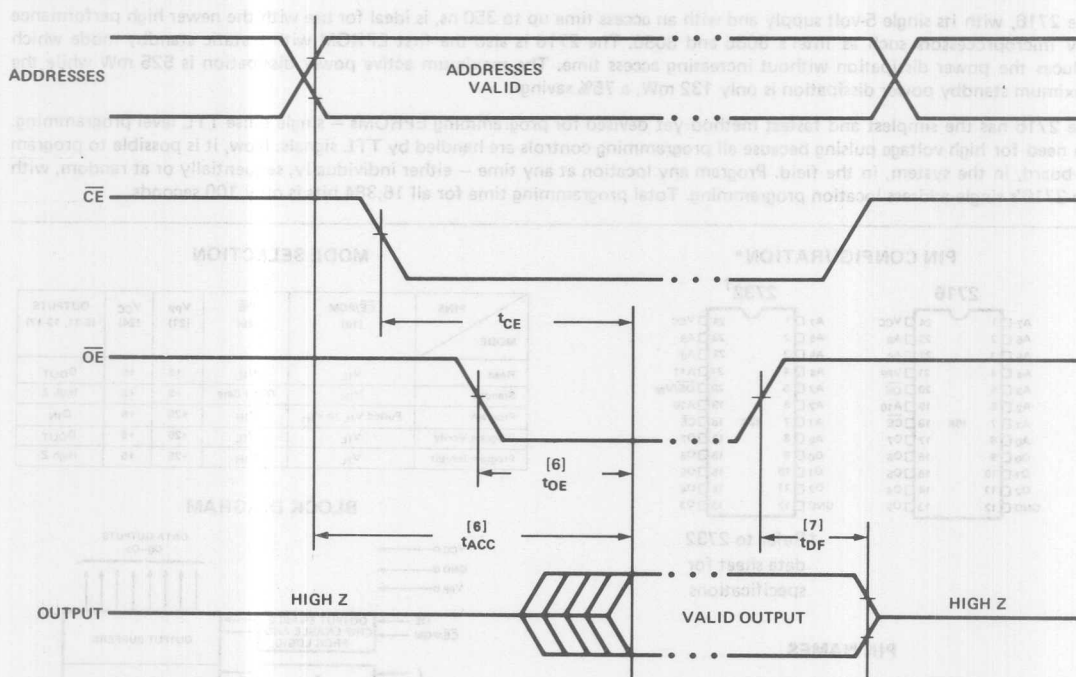
Input Pulse Levels: 0.8V to 2.2V

Timing Measurement Reference Level:

Inputs 1V and 2V

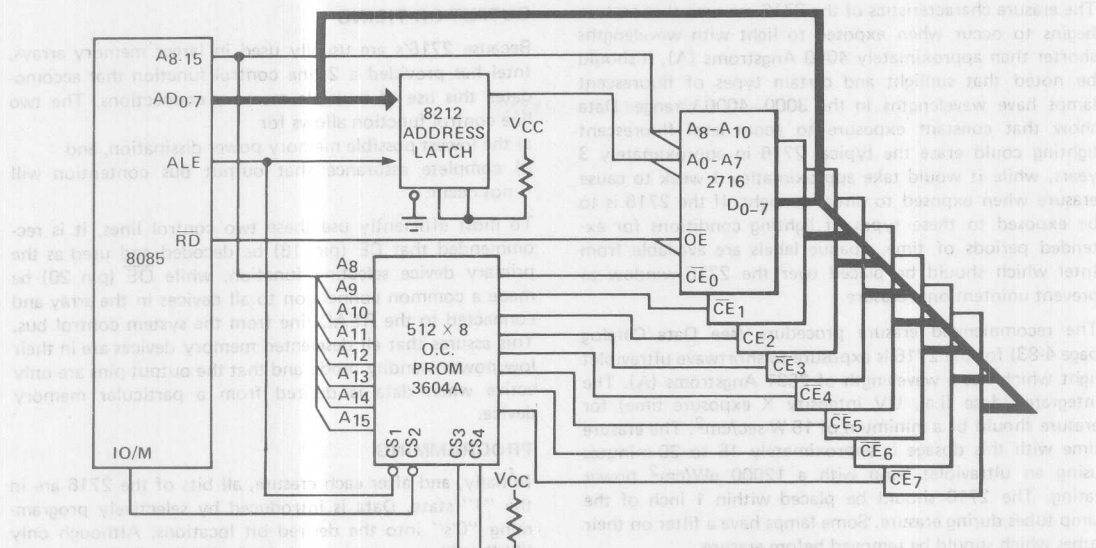
Outputs 0.8V and 2V

## A. C. Waveforms (1)



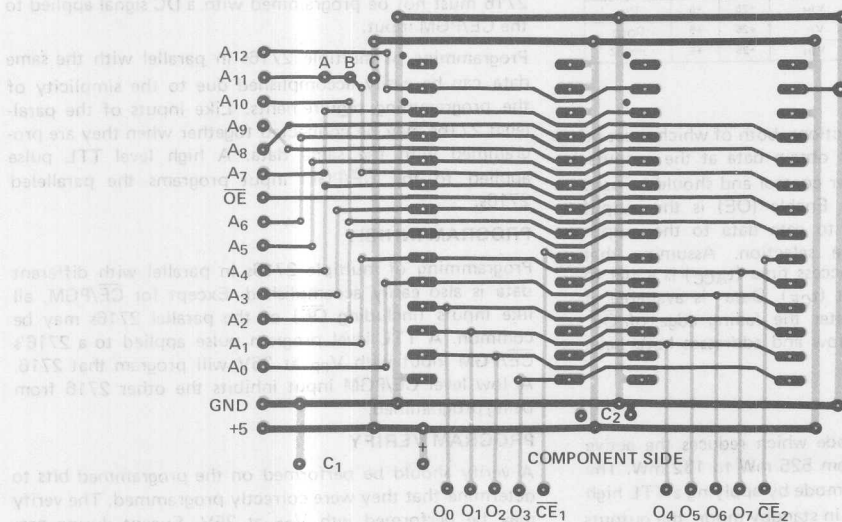
- NOTE: 1. V<sub>CC</sub> must be applied simultaneously or before V<sub>pp</sub> and removed simultaneously or after V<sub>pp</sub>.  
 2. V<sub>pp</sub> may be connected directly to V<sub>CC</sub> except during programming. The supply current would then be the sum of I<sub>CC</sub> and I<sub>pp1</sub>.  
 3. The tolerance of 0.6V allows the use of a driver circuit for switching the V<sub>pp</sub> supply pin from V<sub>CC</sub> in read to 25V for programming.  
 4. Typical values are for T<sub>A</sub> = 25°C and nominal supply voltages.  
 5. This parameter is only sampled and is not 100% tested.  
 6.  $\overline{OE}$  may be delayed up to t<sub>ACC</sub> - t<sub>OE</sub> after the falling edge of  $\overline{CE}$  without impact on t<sub>ACC</sub>.  
 7. t<sub>DF</sub> is specified from  $\overline{OE}$  or  $\overline{CE}$ , whichever occurs first.

## TYPICAL 16K EPROM SYSTEM



- This scheme accomplished by using  $\overline{CE}$  (PD) as the primary decode.  $\overline{OE}$  (CS) is now controlled by previously unused signal. RD now controls data on and off the bus by way of  $\overline{OE}$ .
- A selected 2716 is available for systems which require  $\overline{CE}$  access of less than 450 ns for decode network operation.
- The use of a PROM as a decoder allows for:
  - a) ALE is required for Edge Enabled devices (32K and 64K), and is optional for 2716.
  - b) Compatibility with upward (and downward) memory expansion.
  - c) Easy assignment of ROM memory modules, compatible with PL/M modular software concepts.

### 8K, 16K, 32K, 64K 5V EPROM/ROM FAMILY PRINTED CIRCUIT BOARD LAYOUT



begins to occur when exposed to light with wavelength shorter than approximately 4000 Angstroms (Å). It should be noted that sunlight and certain types of fluorescent lamps have wavelengths in the 3000–4000Å range. Data show that constant exposure to room level fluorescent lighting could erase the typical 2716 in approximately 3 years, while it would take approximately 1 week to cause erasure when exposed to direct sunlight. If the 2716 is to be exposed to these types of lighting conditions for extended periods of time, opaque labels are available from Intel which should be placed over the 2716 window to prevent unintentional erasure.

The recommended erasure procedure (see Data Catalog page 4-83) for the 2716 is exposure to shortwave ultraviolet light which has a wavelength of 2537 Angstroms (Å). The integrated dose (i.e., UV intensity X exposure time) for erasure should be a minimum of 15 W-sec/cm<sup>2</sup>. The erasure time with this dosage is approximately 15 to 20 minutes using an ultraviolet lamp with a 12000 μW/cm<sup>2</sup> power rating. The 2716 should be placed within 1 inch of the lamp tubes during erasure. Some lamps have a filter on their tubes which should be removed before erasure.

## DEVICE OPERATION

The five modes of operation of the 2716 are listed in Table I. It should be noted that all inputs for the five modes are at TTL levels. The power supplies required are a +5V V<sub>CC</sub> and a V<sub>PP</sub>. The V<sub>PP</sub> power supply must be at 25V during the three programming modes, and must be at 5V in the other two modes.

TABLE I. MODE SELECTION

MODE \ PINS	CE/PGM (18)	OE (20)	V <sub>PP</sub> (21)	V <sub>CC</sub> (24)	OUTPUTS (9-11, 13-17)
Read	V <sub>IL</sub>	V <sub>IL</sub>	+5	+5	D <sub>OUT</sub>
Standby	V <sub>IH</sub>	Don't Care	+5	+5	High Z
Program	Pulsed V <sub>IL</sub> to V <sub>IH</sub>	V <sub>IH</sub>	+25	+5	D <sub>IN</sub>
Program Verify	V <sub>IL</sub>	V <sub>IL</sub>	+25	+5	D <sub>OUT</sub>
Program Inhibit	V <sub>IL</sub>	V <sub>IH</sub>	+25	+5	High Z

## READ MODE

The 2716 has two control functions, both of which must be logically satisfied in order to obtain data at the outputs. Chip Enable (CE) is the power control and should be used for device selection. Output Enable (OE) is the output control and should be used to gate data to the output pins, independent of device selection. Assuming that addresses are stable, address access time (t<sub>ACC</sub>) is equal to the delay from CE to output (t<sub>CE</sub>). Data is available at the outputs 120 ns (t<sub>OE</sub>) after the falling edge of OE, assuming that CE has been low and addresses have been stable for at least t<sub>ACC</sub> - t<sub>OE</sub>.

## STANDBY MODE

The 2716 has a standby mode which reduces the active power dissipation by 75%, from 525 mW to 132 mW. The 2716 is placed in the standby mode by applying a TTL high signal to the CE input. When in standby mode, the outputs are in a high impedance state, independent of the OE input.

2716

Because 2716's are usually used in larger memory arrays, Intel has provided a 2 line control function that accommodates this use of multiple memory connections. The two line control function allows for

- the lowest possible memory power dissipation, and
- complete assurance that output bus contention will not occur.

To most efficiently use these two control lines, it is recommended that CE (pin 18) be decoded and used as the primary device selecting function, while OE (pin 20) be made a common connection to all devices in the array and connected to the READ line from the system control bus. This assures that all deselected memory devices are in their low power standby mode and that the output pins are only active when data is desired from a particular memory device.

## PROGRAMMING

Initially, and after each erasure, all bits of the 2716 are in the "1" state. Data is introduced by selectively programming "0's" into the desired bit locations. Although only "0's" will be programmed, both "1's" and "0's" can be presented in the data word. The only way to change a "0" to a "1" is by ultraviolet light erasure.

The 2716 is in the programming mode when the V<sub>PP</sub> power supply is at 25V and OE is at V<sub>IH</sub>. The data to be programmed is applied 8 bits in parallel to the data output pins. The levels required for the address and data inputs are TTL.

When the address and data are stable, a 50 msec, active high, TTL program pulse is applied to the CE/PGM input. A program pulse must be applied at each address location to be programmed. You can program any location at any time — either individually, sequentially, or at random. The program pulse has a maximum width of 55 msec. The 2716 must not be programmed with a DC signal applied to the CE/PGM input.

Programming of multiple 2716s in parallel with the same data can be easily accomplished due to the simplicity of the programming requirements. Like inputs of the paralleled 2716s may be connected together when they are programmed with the same data. A high level TTL pulse applied to the CE/PGM input programs the paralleled 2716s.

## PROGRAM INHIBIT

Programming of multiple 2716s in parallel with different data is also easily accomplished. Except for CE/PGM, all like inputs (including OE) of the parallel 2716s may be common. A TTL level program pulse applied to a 2716's CE/PGM input with V<sub>PP</sub> at 25V will program that 2716. A low level CE/PGM input inhibits the other 2716 from being programmed.

## PROGRAM VERIFY

A verify should be performed on the programmed bits to determine that they were correctly programmed. The verify may be performed with V<sub>PP</sub> at 25V. Except during programming and program verify, V<sub>PP</sub> must be at 5V.

## PROGRAMMING

The programming specifications are described in the Data Catalog PROM/ROM Programming Instructions on Page 4-83.

### Absolute Maximum Ratings\*

Temperature Under Bias	-10°C to +80°C
Storage Temperature	-65°C to +125°C
All Input or Output Voltages with Respect to Ground	+6V to -0.3V
V <sub>PP</sub> Supply Voltage with Respect to Ground During Program	+26.5V to -0.3V

\*COMMENT: Stresses above those listed under "Absolute Maximum Ratings" may cause permanent damage to the device. This is a stress rating only and functional operation of the device at these or any other conditions above those indicated in the operational sections of this specification is not implied. Exposure to absolute maximum rating conditions for extended periods may affect device reliability.

### DC and AC Operating Conditions During Read

	2716	2716-1	2716-2
Temperature Range	0°C - 70°C	0°C - 70°C	0°C - 70°C
V <sub>CC</sub> Power Supply <sup>[1,2]</sup>	5V ± 5%	5V ± 10%	5V ± 5%
V <sub>PP</sub> Power Supply <sup>[2]</sup>	V <sub>CC</sub> ± 0.6V <sup>[3]</sup>	V <sub>CC</sub> ± 0.6V <sup>[3]</sup>	V <sub>CC</sub> ± 0.6V <sup>[3]</sup>

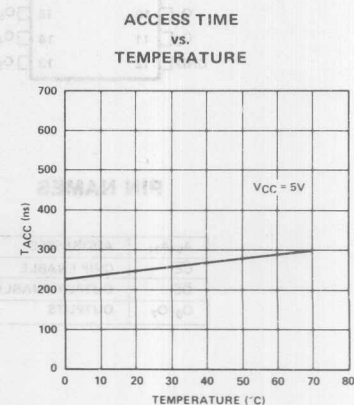
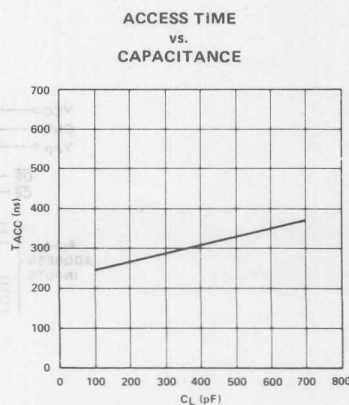
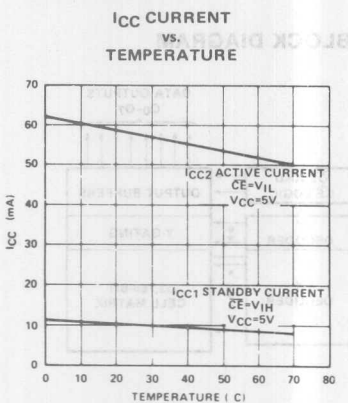
## READ OPERATION

### D.C. and Operating Characteristics

Symbol	Parameter	Limits			Unit	Conditions
		Min.	Typ. <sup>[4]</sup>	Max.		
I <sub>LI</sub>	Input Load Current			10	μA	V <sub>IN</sub> = 5.25V
I <sub>LO</sub>	Output Leakage Current			10	μA	V <sub>OUT</sub> = 5.25V
I <sub>PP1</sub> <sup>[2]</sup>	V <sub>PP</sub> Current			5	mA	V <sub>PP</sub> = 5.85V
I <sub>CC1</sub> <sup>[2]</sup>	V <sub>CC</sub> Current (Standby)		10	25	mA	$\overline{CE} = V_{IH}$ , $\overline{OE} = V_{IL}$
I <sub>CC2</sub> <sup>[2]</sup>	V <sub>CC</sub> Current (Active)		57	100	mA	$\overline{OE} = \overline{CE} = V_{IL}$
V <sub>IL</sub>	Input Low Voltage	-0.1		0.8	V	
V <sub>IH</sub>	Input High Voltage	2.0		V <sub>CC</sub> +1	V	
V <sub>OL</sub>	Output Low Voltage			0.45	V	I <sub>OL</sub> = 2.1 mA
V <sub>OH</sub>	Output High Voltage	2.4			V	I <sub>OH</sub> = -400 μA

- NOTES: 1. V<sub>CC</sub> must be applied simultaneously or before V<sub>PP</sub> and removed simultaneously or after V<sub>PP</sub>.  
 2. V<sub>PP</sub> may be connected directly to V<sub>CC</sub> except during programming. The supply current would then be the sum of I<sub>CC</sub> and I<sub>PP1</sub>.  
 3. The tolerance of 0.6V allows the use of a driver circuit for switching the V<sub>PP</sub> supply pin from V<sub>CC</sub> in read to 25V for programming.  
 4. Typical values are for T<sub>A</sub> = 25°C and nominal supply voltages.  
 5. This parameter is only sampled and is not 100% tested.

### Typical Characteristics



# 2732

## 32K (4K x 8) UV ERASABLE PROM

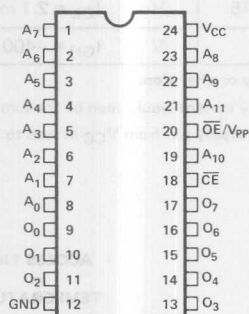
- Single +5V  $\pm$  5% Power Supply
- Output Enable for MCS-85™ and MCS-86™ Compatibility
- Fast Access Time: 450ns Max.
- Low Power Dissipation:  
160mA Max. Active Current  
25mA Max. Standby Current
- Pin Compatible to Intel® 2716 EPROM and 2332/2364 ROMs
- Completely Static
- Simple Programming Requirements
  - Single Location Programming
  - Programs with One 50ms Pulse
- Three-State Output for Direct Bus Interface

The Intel® 2732 is a 32,768-bit ultraviolet erasable and electrically programmable read-only memory (EPROM). The 2732 operates from a single 5-volt power supply, has a standby mode, and features an output enable control. For production, the pin compatible 2332 and 2364 ROMs are available. The total programming time for all bits is three and a half minutes. All these features make designing with the 2732 in microcomputer systems faster, easier, and more economical.

An important 2732 feature is the separate output control, Output Enable ( $\overline{OE}$ ), from the Chip Enable control ( $\overline{CE}$ ). The  $\overline{OE}$  control eliminates bus contention in multiple bus microprocessor systems. Intel's Application Note AP-30 describes the microprocessor system implementation of the  $\overline{OE}$  and  $\overline{CE}$  controls on Intel's 2716 and 2732 EPROMs. AP-30 is available from Intel's Literature Department.

The 2732 has a standby mode which reduces the power dissipation without increasing access time. The maximum active current is 160mA, while the maximum standby current is only 25mA, an 85% savings. The standby mode is achieved by applying a TTL-high signal to the  $\overline{CE}$  input.

### PIN CONFIGURATION



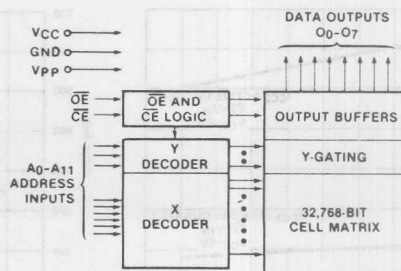
### PIN NAMES

A <sub>0</sub> -A <sub>11</sub>	ADDRESSES
$\overline{CE}$	CHIP ENABLE
$\overline{OE}$	OUTPUT ENABLE
O <sub>0</sub> -O <sub>7</sub>	OUTPUTS

### MODE SELECTION

PINS MODE	$\overline{CE}$ (18)	$\overline{OE}/V_{pp}$ (20)	V <sub>CC</sub> (24)	OUTPUTS (9-11,13-17)
Read	V <sub>IL</sub>	V <sub>IL</sub>	+5	D <sub>OUT</sub>
Standby	V <sub>IH</sub>	Don't Care	+5	High Z
Program	V <sub>IL</sub>	V <sub>IHP</sub>	+5	D <sub>IN</sub>
Program Verify	V <sub>IL</sub>	V <sub>IL</sub>	+5	D <sub>OUT</sub>
Program Inhibit	V <sub>IH</sub>	V <sub>IHP</sub>	+5	High Z

### BLOCK DIAGRAM



# 2758\*

## 8K (1K × 8) UV ERASABLE LOW POWER PROM

- Single +5V Power Supply
- Simple Programming Requirements  
Single Location Programming  
Programs with One 50 ms Pulse
- Low Power Dissipation  
525 mW Max. Active Power  
132 mW Max. Standby Power
- Fast Access Time: 450 ns Max. in  
Active and Standby Power Modes
- Inputs and Outputs TTL Compatible  
during Read and Program
- Completely Static
- Three-State Outputs for OR-Ties

The Intel® 2758 is a 8192-bit ultraviolet erasable and electrically programmable read-only memory (EPROM). The 2758 operates from a single 5-volt power supply, has a static standby mode, and features fast single address location programming. It makes designing with EPROMs faster, easier and more economical. The total programming time for all 8192 bits is 50 seconds.

The 2758 has a static standby mode which reduces the power dissipation without increasing access time. The maximum active power dissipation is 525 mW, while the maximum standby power dissipation is only 132 mW, a 75% savings. Power-down is achieved by applying a TTL-high signal to the  $\overline{CE}$  input.

A 2758 system may be designed for total upwards compatibility with Intel's 16K 2716 EPROM (see Applications Note 30). The 2758 maintains the simplest and fastest method yet devised for programming EPROMs — single pulse TTL-level programming. There is no need for high voltage pulsing because all programming controls are handled by TTL signals. Now it is possible to program on-board, in the system, in the field. Program any location at any time — either individually, sequentially, or at random, with the single address location programming.

### PIN CONFIGURATION\*

A7	1	24	VCC
A6	2	23	A8
A5	3	22	A9
A4	4	21	Vpp
A3	5	20	$\overline{OE}$
A2	6	19	A <sub>R</sub>
A1	7	18	$\overline{CE}$
A0	8	17	O <sub>7</sub>
O <sub>0</sub>	9	16	O <sub>6</sub>
O <sub>1</sub>	10	15	O <sub>5</sub>
O <sub>2</sub>	11	14	O <sub>4</sub>
GND	12	13	O <sub>3</sub>

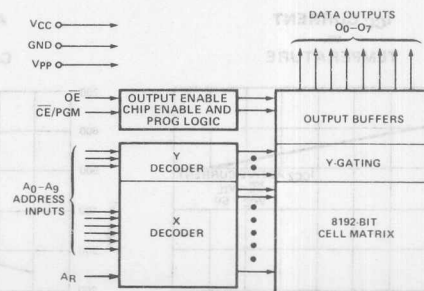
### MODE SELECTION

PINS	$\overline{CE}/\text{PGM}$ (18)	A <sub>R</sub> (19)	$\overline{OE}$ (20)	V <sub>pp</sub> (21)	V <sub>CC</sub> (24)	OUTPUTS (9-11, 13-17)
Read	V <sub>IL</sub>	V <sub>IL</sub>	V <sub>IL</sub>	+5	+5	D <sub>OUT</sub>
Standby	V <sub>IH</sub>	V <sub>IL</sub>	Don't Care	+5	+5	High Z
Program	Pulsed V <sub>IL</sub> to V <sub>IH</sub>	V <sub>IL</sub>	V <sub>IH</sub>	+25	+5	D <sub>IN</sub>
Program Verify	V <sub>IL</sub>	V <sub>IL</sub>	V <sub>IL</sub>	+25	+5	D <sub>OUT</sub>
Program Inhibit	V <sub>IL</sub>	V <sub>IL</sub>	V <sub>IH</sub>	+25	+5	High Z

### BLOCK DIAGRAM

#### PIN NAMES

A <sub>0</sub> –A <sub>9</sub>	ADDRESSES
$\overline{CE}/\text{PGM}$	CHIP ENABLE/PROGRAM
$\overline{OE}$	OUTPUT ENABLE
O <sub>0</sub> –O <sub>7</sub>	OUTPUTS
A <sub>R</sub>	SELECT REFERENCE INPUT LEVEL



\*Pin 18 and pin 20 have been renamed to conform with the entire family of 16K, 32K, and 64K EPROMs and ROMs. The die, fabrication process, and specifications remain the same and are totally unaffected by this change.

## Absolute Maximum Ratings\*

Temperature Under Bias . . . . .	-10°C to +80°C
Storage Temperature . . . . .	-65°C to +125°C
All Input or Output Voltages with Respect to Ground . . . . .	+6V to -0.3V
V <sub>PP</sub> Supply Voltage with Respect to Ground During Programming . . . . .	+26.5V to -0.3V

\*COMMENT: Stresses above those listed under "Absolute Maximum Ratings" may cause permanent damage to the device. This is a stress rating only and functional operation of the device at these or any other conditions above those indicated in the operational sections of this specification is not implied. Exposure to absolute maximum rating conditions for extended periods may affect device reliability.

## READ OPERATION

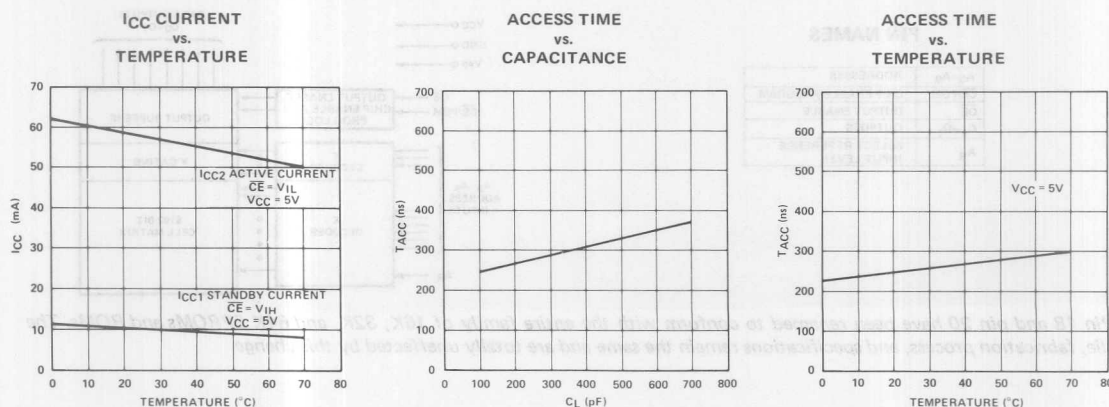
### D.C. and Operating Characteristics

T<sub>A</sub> = 0°C to 70°C, V<sub>CC</sub><sup>[1,2]</sup> = +5V ±5%, V<sub>PP</sub><sup>[2]</sup> = V<sub>CC</sub> ±0.6V<sup>[3]</sup>

Symbol	Parameter	Limits			Unit	Conditions
		Min.	Typ. <sup>[4]</sup>	Max.		
I <sub>LI</sub>	Input Load Current			10	μA	V <sub>IN</sub> = 5.25V
I <sub>LO</sub>	Output Leakage Current			10	μA	V <sub>OUT</sub> = 5.25V
I <sub>PP1</sub> <sup>[2]</sup>	V <sub>PP</sub> Current			5	mA	V <sub>PP</sub> = 5.85V
I <sub>CC1</sub> <sup>[2]</sup>	V <sub>CC</sub> Current (Standby)		10	25	mA	$\overline{CE} = V_{IH}, \overline{OE} = V_{IL}$
I <sub>CC2</sub> <sup>[2]</sup>	V <sub>CC</sub> Current (Active)		57	100	mA	$\overline{OE} = \overline{CE} = V_{IL}$
A <sub>R</sub> <sup>[5]</sup>	Select Reference Input Level	-0.1		0.8	V	I <sub>IN</sub> = 10 μA
V <sub>IL</sub>	Input Low Voltage	-0.1		0.8	V	
V <sub>IH</sub>	Input High Voltage	2.0		V <sub>CC</sub> + 1	V	
V <sub>OL</sub>	Output Low Voltage			0.45	V	I <sub>OL</sub> = 2.1 mA
V <sub>OH</sub>	Output High Voltage	2.4			V	I <sub>OH</sub> = -400 μA

- NOTES:
1. V<sub>CC</sub> must be applied simultaneously or before V<sub>PP</sub> and removed simultaneously or after V<sub>PP</sub>.
  2. V<sub>PP</sub> may be connected directly to V<sub>CC</sub> except during programming. The supply current would then be the sum of I<sub>CC</sub> and I<sub>PP1</sub>.
  3. The tolerance of 0.6V allows the use of a driver circuit for switching the V<sub>PP</sub> supply pin from V<sub>CC</sub> in read to 25V for programming.
  4. Typical values are for T<sub>A</sub> = 25°C and nominal supply voltages.
  5. A<sub>R</sub> is a reference voltage level which requires an input current of only 10 μA. The 2758 S1865 is also available which has a reference voltage level of V<sub>IH</sub> instead of V<sub>IL</sub>.

## Typical Characteristics



## A.C. Characteristics

 $T_A = 0^\circ\text{C to } 70^\circ\text{C}$ ,  $V_{CC}^{[1]} = +5\text{V} \pm 5\%$ ,  $V_{PP}^{[2]} = V_{CC} \pm 0.6\text{V}^{[3]}$ 

Symbol	Parameter	Limits			Unit	Test Conditions
		Min.	Typ. <sup>[4]</sup>	Max.		
$t_{ACC}$	Address to Output Delay		250	450	ns	$\overline{CE} = \overline{OE} = V_{IL}$
$t_{CE}$	$\overline{CE}$ to Output Delay		280	450	ns	$\overline{OE} = V_{IL}$
$t_{OE}$	Output Enable to Output Delay			120	ns	$\overline{CE} = V_{IL}$
$t_{DF}$	Output Enable High to Output Float	0		100	ns	$\overline{CE} = V_{IL}$
$t_{OH}$	Address to Output Hold	0			ns	$\overline{CE} = \overline{OE} = V_{IL}$

**Capacitance**<sup>[5]</sup>  $T_A = 25^\circ\text{C}$ ,  $f = 1\text{ MHz}$ 

Symbol	Parameter	Typ.	Max.	Unit	Conditions
$C_{IN}$	Input Capacitance	4	6	pF	$V_{IN} = 0\text{V}$
$C_{OUT}$	Output Capacitance	8	12	pF	$V_{OUT} = 0\text{V}$

NOTE: Please refer to page 2 for notes.

## A.C. Test Conditions:

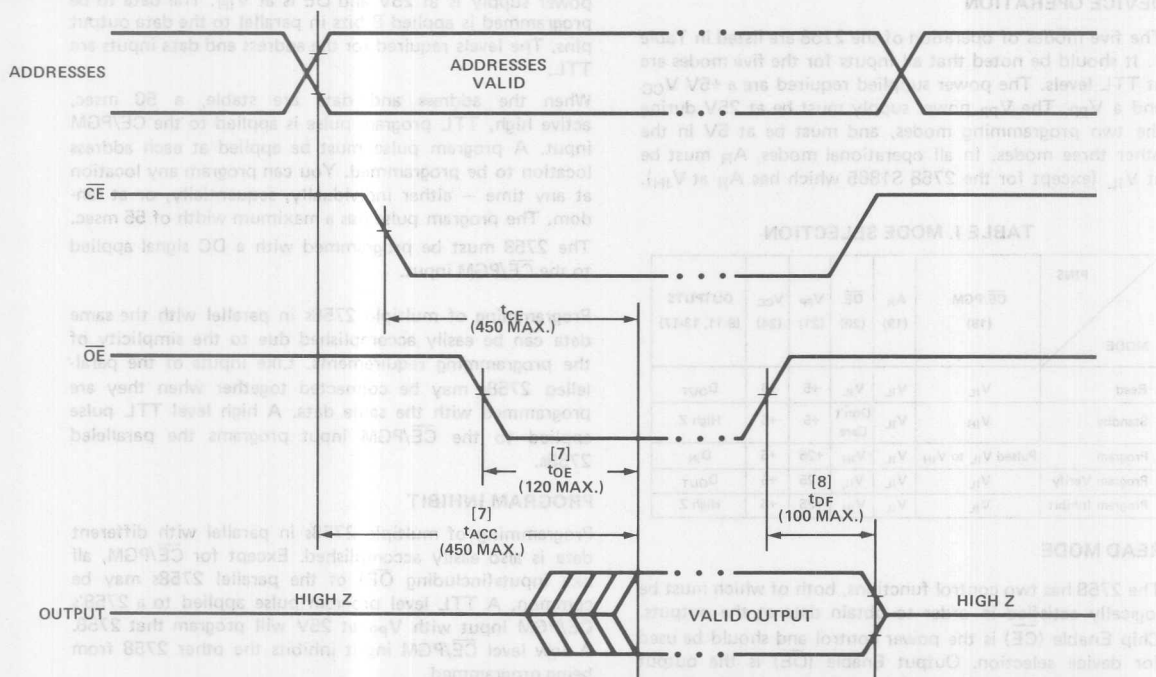
Output Load: 1 TTL gate and  $C_L = 100\text{ pF}$ Input Rise and Fall Times:  $\leq 20\text{ ns}$ 

Input Pulse Levels: 0.8V to 2.2V

Timing Measurement Reference Level:

Inputs 1V and 2V

Outputs 0.8V and 2V

A.C. Waveforms<sup>[6]</sup>

- NOTES:
- $V_{CC}$  must be applied simultaneously or before  $V_{PP}$  and removed simultaneously or after  $V_{PP}$ .
  - $V_{PP}$  may be connected directly to  $V_{CC}$  except during programming. The supply current would then be the sum of  $I_{CC}$  and  $I_{PP1}$ .
  - The tolerance of 0.6V allows the use of a driver circuit for switching the  $V_{PP}$  supply pin from  $V_{CC}$  in read to 25V for programming.
  - Typical values are for  $T_A = 25^\circ\text{C}$  and nominal supply voltages.
  - This parameter is only sampled and is not 100% tested.
  - All times shown in parentheses are minimum times and are nsec unless otherwise specified.
  - $\overline{OE}$  may be delayed up to 330ns after the falling edge of  $\overline{CE}$  without impact on  $t_{ACC}$ .
  - $t_{DF}$  is specified from  $\overline{OE}$  or  $\overline{CE}$ , whichever occurs first.

## ERASURE CHARACTERISTICS

The erasure characteristics of the 2758 are such that erasure begins to occur when exposed to light with wavelengths shorter than approximately 4000 Angstroms (Å). It should be noted that sunlight and certain types of fluorescent lamps have wavelengths in the 3000–4000Å range. Data show that constant exposure to room level fluorescent lighting could erase the typical 2758 in approximately 3 years, while it would take approximately 1 week to cause erasure when exposed to direct sunlight. If the 2758 is to be exposed to these types of lighting conditions for extended periods of time, opaque labels are available from Intel which should be placed over the 2758 window to prevent unintentional erasure.

The recommended erasure procedure (see Data Catalog page 4-83) for the 2758 is exposure to shortwave ultraviolet light which has a wavelength of 2537 Angstroms (Å). The integrated dose (i.e., UV intensity X exposure time) for erasure should be a minimum of 15 W-sec/cm<sup>2</sup>. The erasure time with this dosage is approximately 15 to 20 minutes using an ultraviolet lamp with 12,000 μW/cm<sup>2</sup> power rating. The 2758 should be placed within 1 inch of the lamp tubes during erasure. Some lamps have a filter on their tubes which should be removed before erasure.

## DEVICE OPERATION

The five modes of operation of the 2758 are listed in Table 1. It should be noted that all inputs for the five modes are at TTL levels. The power supplied required are a +5V V<sub>CC</sub> and a V<sub>PP</sub>. The V<sub>PP</sub> power supply must be at 25V during the two programming modes, and must be at 5V in the other three modes. In all operational modes, A<sub>R</sub> must be at V<sub>IL</sub> (except for the 2758 S1865 which has A<sub>R</sub> at V<sub>IH</sub>).

TABLE 1. MODE SELECTION

PINS MODE	$\overline{CE}/PGM$ (18)	A <sub>R</sub> (19)	$\overline{OE}$ (20)	V <sub>PP</sub> (21)	V <sub>CC</sub> (24)	OUTPUTS (9-11, 13-17)
Read	V <sub>IL</sub>	V <sub>IL</sub>	V <sub>IL</sub>	+5	+5	D <sub>OUT</sub>
Standby	V <sub>IH</sub>	V <sub>IL</sub>	Don't Care	+5	+5	High Z
Program	Pulsed V <sub>IL</sub> to V <sub>IH</sub>	V <sub>IL</sub>	V <sub>IH</sub>	+25	+5	D <sub>IN</sub>
Program Verify	V <sub>IL</sub>	V <sub>IL</sub>	V <sub>IL</sub>	+25	+5	D <sub>OUT</sub>
Program Inhibit	V <sub>IL</sub>	V <sub>IL</sub>	V <sub>IH</sub>	+25	+5	High Z

## READ MODE

The 2758 has two control functions, both of which must be logically satisfied in order to obtain data at the outputs. Chip Enable (CE) is the power control and should be used for device selection. Output Enable ( $\overline{OE}$ ) is the output control and should be used to gate data to the output pins, independent of device selection. Assuming that addresses are stable, address access time (t<sub>ACC</sub>) is equal to the delay from CE to output (t<sub>CE</sub>). Data is available at the outputs 120 ns (t<sub>OE</sub>) after the falling edge of  $\overline{OE}$ , assuming that CE has been low and addresses have been stable for at least t<sub>ACC</sub> – t<sub>OE</sub>.

## STANDBY MODE

The 2758 has a standby mode which reduces the active power dissipation by 75%, from 525 mW to 132 mW. The 2758 is placed in the standby mode by applying a TTL high signal to CE input. When in standby mode, the outputs are in a high impedance state, independent of the OE input.

## OUTPUT DESELECTION

The outputs of two or more 2758s may be OR-tied together on the same data bus. Only one 2758 should have its output selected ( $\overline{OE}$  low) to prevent data bus contention between 2758s in this configuration. The outputs of the other 2758s should be deselected by raising the  $\overline{OE}$  input to a TTL high level.

## PROGRAMMING

Initially, and after each erasure, all bits of the 2758 are in the "1" state. Data is introduced by selectively programming "0's" into the desired bit locations. Although only "0's" will be programmed, both "1's" and "0's" can be presented in the data word. The only way to change a "0" to a "1" is by ultraviolet light erasure.

The 2758 is in the programming mode when the V<sub>PP</sub> power supply is at 25V and  $\overline{OE}$  is at V<sub>IH</sub>. The data to be programmed is applied 8 bits in parallel to the data output pins. The levels required for the address and data inputs are TTL.

When the address and data are stable, a 50 msec, active high, TTL program pulse is applied to the CE/PGM input. A program pulse must be applied at each address location to be programmed. You can program any location at any time — either individually, sequentially, or at random. The program pulse has a maximum width of 55 msec. The 2758 must be programmed with a DC signal applied to the CE/PGM input.

Programming of multiple 2758s in parallel with the same data can be easily accomplished due to the simplicity of the programming requirements. Like inputs of the paralleled 2758s may be connected together when they are programmed with the same data. A high level TTL pulse applied to the CE/PGM input programs the paralleled 2758s.

## PROGRAM INHIBIT

Programming of multiple 2758s in parallel with different data is also easily accomplished. Except for CE/PGM, all like inputs (including  $\overline{OE}$ ) of the parallel 2758s may be common. A TTL level program pulse applied to a 2758's CE/PGM input with V<sub>PP</sub> at 25V will program that 2758. A low level CE/PGM input inhibits the other 2758 from being programmed.

## PROGRAM VERIFY

A verify should be performed on the programmed bits to determine that they were correctly programmed. The verify may be performed with V<sub>PP</sub> at 25V. Except during programming and program verify, V<sub>PP</sub> must be at 5V.



## 3604A, 3624A FAMILY 4K (512 × 8) HIGH-SPEED PROM

	3604A-2 3624A-2	3604A 3624A	3604AL
Max. $T_A$ (ns)	60	70	90
Max. $I_{CC}$ (mA)	170	170	130/25*

\*Standby Current When The Chip is Deselected.

- Fast Access Time  
--60ns Max (3604A-2, 3624A-2)
- Low Standby Power Dissipation  
(3604AL) --32 $\mu$ W/Bit Max
- Open Collector (3604A)  
or Three State (3624A)  
Outputs
- Four Chip Select Inputs  
For Easy Memory  
Expansion
- Polycrystalline Silicon Fuse  
For Higher Reliability
- Hermetic 24 Pin DIP

The Intel® 3604A/3624A are 4096-bit bipolar PROMs organized as 512 words by 8 bits. The fast second generation 3604A/3624A replaces its Intel predecessor, the 3604/3624. Higher speed PROMs, the 3604A-2/3624A-2, are now available at 60 ns. All 3604A/3624A specifications, except programming, are the same as or better than the 3604/3624. Once programmed, the 3604A/3624A are interchangeable with the 3604/3624.

The PROMs are manufactured with all outputs initially logically high. Logic low levels can be electrically programmed in selected bit locations. Both open collector and three-state outputs are available. Low standby power dissipation can be achieved with the 3604AL. The standby power dissipation is approximately 20% of the active power dissipation. The 3604A/3624A are available in a hermetic 24-pin dual in-line package. These PROMs are manufactured with the time-proven polycrystalline silicon fuse technology.

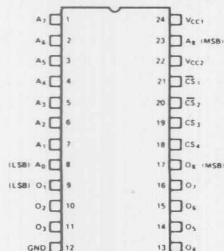
Mode/Pin Connection	Pin 22	Pin 24
READ: 3604A, 3604A-2 3624A, 3624A-2	No Connect or 5V	5V
3604AL	+5V	Must be Left Open
PROGRAM: 3604A, 3604A-2 3624A, 3624A-2	Pulsed 12.5V	Pulsed 12.5V
3604AL	Pulsed 12.5V	Pulsed 12.5V
STANDBY: 3604AL	Power dissipation is automatically reduced whenever the 3604AL is deselected.	

### PIN NAMES

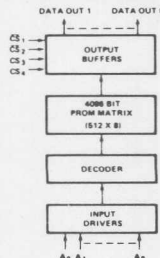
$A_0$ – $A_8$	ADDRESS INPUTS
$CS_1$ – $CS_2$	CHIP SELECT INPUTS [1]
$CS_3$ – $CS_4$	
$O_1$ – $O_8$	DATA OUTPUTS

[1] To select the PROM  $\overline{CS}_1 = \overline{CS}_2 = 0$   
and  $CS_3 = CS_4 = 1$ .

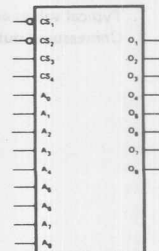
### PIN CONFIGURATION



### BLOCK DIAGRAM



### LOGIC SYMBOL



## PROGRAMMING

The programming specifications are described in the Data Catalog PROM/ROM Programming Instructions on page 4-89.

## Absolute Maximum Ratings\*

Temperature Under Bias	-65°C to +125°C
Storage Temperature	-65°C to +160°C
Output or Supply Voltages	-0.5V to 7 Volts
All Input Voltages	-1.6 to 5.5V
Output Currents	100mA

## \*COMMENT

Stresses above those listed under "Absolute Maximum Rating" may cause permanent damage to the device. This is a stress rating only and functional operation of the device at these or at any other condition above those indicated in the operational sections of this specification is not implied.

D. C. Characteristics: All Limits Apply for  $V_{CC} = +5.0V \pm 5\%$ ,  $T_A = 0^\circ C$  to  $+75^\circ C$ 

Symbol	Parameter	Limits			Unit	Test Conditions
		Min.	Typ. <sup>[1]</sup>	Max.		
$I_{FA}$	Address Input Load Current		-0.05	-0.25	mA	$V_{CC} = 5.25V$ , $V_A = 0.45V$
$I_{FS}$	Chip Select Input Load Current		-0.05	-0.25	mA	$V_{CC} = 5.25V$ , $V_S = 0.45V$
$I_{RA}$	Address Input Leakage Current			40	$\mu A$	$V_{CC} = 5.25V$ , $V_A = 5.25V$
$I_{RS}$	Chip Select Input Leakage Current			40	$\mu A$	$V_{CC} = 5.25V$ , $V_S = 5.25V$
$V_{CA}$	Address Input Clamp Voltage		-0.9	-1.5	V	$V_{CC} = 4.75V$ , $I_A = -10$ mA
$V_{CS}$	Chip Select Input Clamp Voltage		-0.9	-1.5	V	$V_{CC} = 4.75V$ , $I_S = -10$ mA
$V_{OL}$	Output Low Voltage		0.3	0.45	V	$V_{CC} = 4.75V$ , $I_{OL} = 15$ mA
$I_{CEX}$	Output Leakage Current			100	$\mu A$	$V_{CC} = 5.25V$ , $V_{CE} = 5.25V$
$I_{CC1}$	Power Supply Current (3604A, 3604A-2, 3624A, and 3624A-2)		130	170	mA	$V_{CC1} = 5.25V$ , $V_{A0} \rightarrow V_{A8} = 0V$ , $\overline{CS}_1 = \overline{CS}_2 = 0V$ , $CS_3 = CS_4 = 5.25V$
$I_{CC2}$	Power Supply Current (3604AL)					$V_{CC2} = 5.25V$ , $V_{CC1} = \text{Open}$
	Active		100	130	mA	$\overline{CS}_1 = \overline{CS}_2 = 0.45V$ , $CS_3 = CS_4 = 2.4V$
	Standby		15	25	mA	$\overline{CS}_1 = \overline{CS}_2 = 2.5V$
$V_{IL}$	Input "Low" Voltage			0.85	V	$V_{CC} = 5.0V$
$V_{IH}$	Input "High" Voltage	2.0			V	$V_{CC} = 5.0V$

## 3624A FAMILY ONLY

Symbol	Parameter	Min.	Typ. <sup>[1]</sup>	Max.	Unit	Test Conditions
$I_{OL}$	Output Leakage for High Impedance Stage			100	$\mu A$	$V_O = 5.25V$ or $0.45V$ , $V_{CC} = 5.25V$ , $\overline{CS}_1 = \overline{CS}_2 = 2.4V$
$I_{SC}^{[2]}$	Output Short Circuit Current	-20	-25	-70	mA	$V_O = 0V$ , $V_{CC} = 4.75V$
$V_{OH}$	Output High Voltage	2.4			V	$I_{OH} = -2.4$ mA, $V_{CC} = 4.75V$

NOTES: 1. Typical values are at  $25^\circ C$  and at nominal voltage.  
2. Unmeasured outputs are open during this test.

## A. C. Characteristics $V_{CC} = +5V \pm 5\%$ , $T_A = 0^\circ C$ to $+75^\circ C$

SYMBOL	PARAMETER	MAXIMUM LIMITS (ns)			UNIT	TEST CONDITIONS
		3604A-2 3624A-2	3604A 3624A	3604AL		
$t_{A++}$ , $t_{A--}$ $t_{A+-}$ , $t_{A-+}$	Address to Output Delay	60	70	90	ns	$\overline{CS}_1 = \overline{CS}_2 = V_{IL}$ and $CS_3 = CS_4 = V_{IH}$ to Select the PROM
$t_{S++}$	Chip Select to Output Delay	30	30	30	ns	
$t_{S--}$	Chip Select to Output Delay	30	30	120	ns	

## Capacitance <sup>(1)</sup> $T_A = 25^\circ C$ , $f = 1$ MHz

SYMBOL	PARAMETER	LIMITS		UNIT	TEST CONDITIONS
		TYP.	MAX.		
$C_{INA}$	Address Input Capacitance	4	10	pF	$V_{CC} = 5V$ $V_{IN} = 2.5V$
$C_{INS}$	Chip-Select Input Capacitance	6	10	pF	$V_{CC} = 5V$ $V_{IN} = 2.5V$
$C_{OUT}$	Output Capacitance	7	15	pF	$V_{CC} = 5V$ $V_{OUT} = 2.5V$

NOTE 1: This parameter is only periodically sampled and is not 100% tested.

## Switching Characteristics

### Conditions of Test:

Input pulse amplitudes - 2.5V

Input pulse rise and fall times of

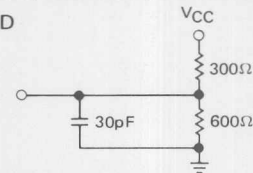
5 nanoseconds between 1 volt and 2 volts

Speed measurements are made at 1.5 volt levels

Output loading is 15 mA and 30 pF

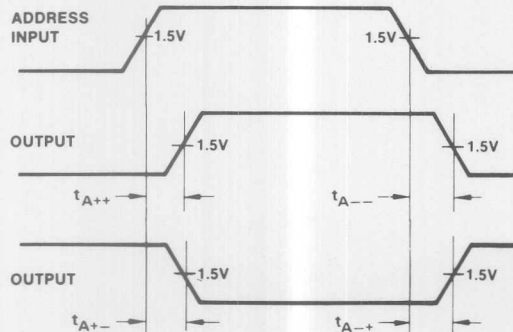
Frequency of test - 2.5 MHz

15 mA TEST LOAD

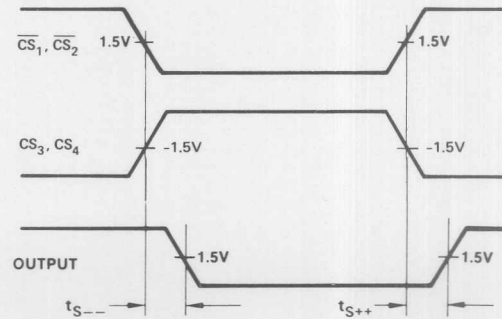


## Waveforms

### ADDRESS TO OUTPUT DELAY



### CHIP SELECT TO OUTPUT DELAY





# 3605A, 3625A 4K (1K × 4) PROM

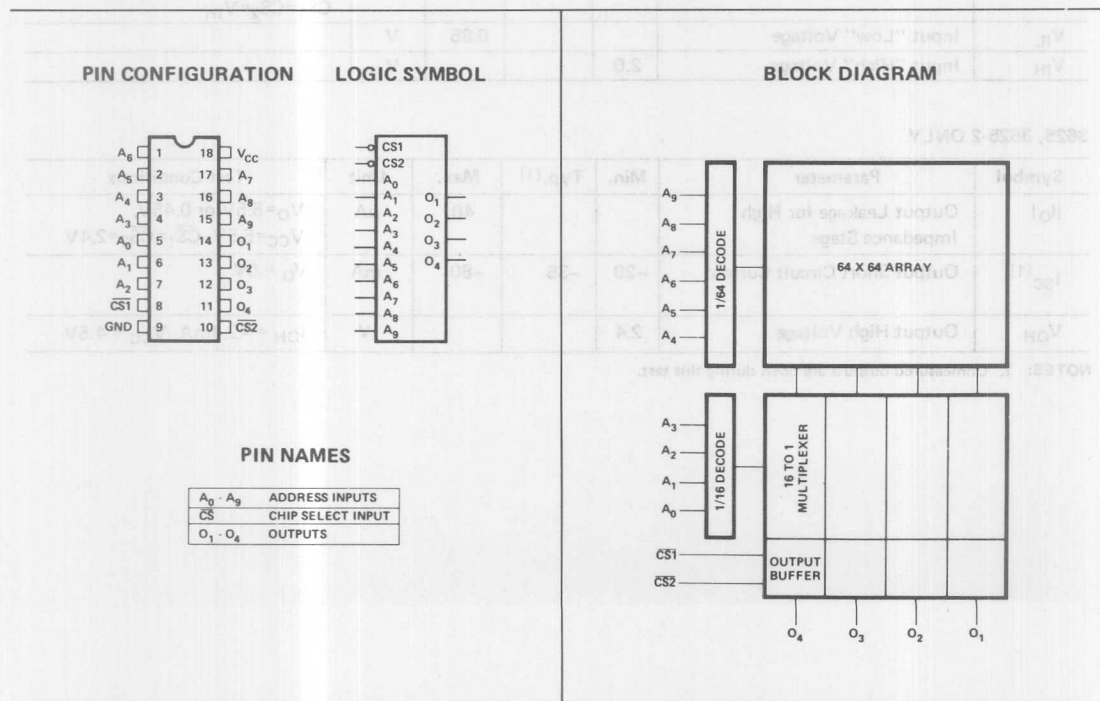
3605A-1, 3625A-1	50 ns Max.
3605A, 3625A	60 ns Max.

- $\pm 10\%$  Power Supply Tolerance
- Fast Access Time: 40 ns Typically
- Lower Power Dissipation: 0.14 mW/Bit Typically
- Simple Memory Expansion Two Chip Select Inputs
- Open Collector (3605A) and Three-State (3625A) Outputs
- Polycrystalline Silicon Fuse for Higher Reliability
- Hermetic 18-Pin DIP

The Intel® 3605A and 3625A families are high density, 4096-bit bipolar PROMs organized as 1024 words by 4 bits. The 1024 by 4 organization gives ideal word or bit modularity for memory array expansion. The 3605A has open collector outputs and the 3625A has three-state outputs. The 3605A and 3625A are fully specified over the 0°C to 75°C temperature range with  $\pm 10\%$  power supply variation. Maximum access times of 50 ns (3605A-2/3625A-2) and 60 ns (3605A/3625A) are available at a typical power dissipation of 0.14 mW/bit.

The 3605A/3625A are packaged in an 18-pin dual in-line hermetic package with 300 milli-inch centers. Thus, twice the bit density can be achieved with the 3605A/3625A in the same memory board areas as 512 by 8-bit PROMs in 24-pin packages.

The highly reliable polycrystalline silicon fuse technology is used in the manufacturing of the 3605A and 3625A families. All outputs are initially a logical high and logic low levels can be electrically programmed in selected bit locations.



## Absolute Maximum Ratings\*

Temperature Under Bias	-65°C to +125°C
Storage Temperature	-65°C to +160°C
Output or Supply Voltages	-0.5V to 7 Volts
All Input Voltages	-1V to 5.5V
Output Currents	100mA

### \*COMMENT

Stresses above those listed under "Absolute Maximum Rating" may cause permanent damage to the device. This is a stress rating only and functional operation of the device at these or at any other condition above those indicated in the operational sections of this specification is not implied.

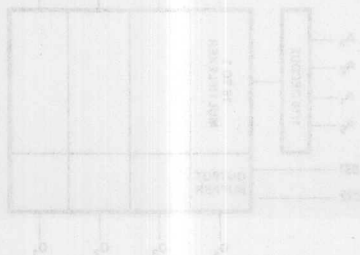
## D. C. Characteristics: All Limits Apply for $V_{CC} = +5.0V \pm 10\%$ , $T_A = 0^\circ C$ to $+75^\circ C$

Symbol	Parameter	Limits				Test Conditions
		Min.	Typ. <sup>[1]</sup>	Max.	Unit	
$I_{FA}$	Address Input Load Current		-0.05	-0.25	mA	$V_{CC}=5.5V$ , $V_A=0.45V$
$I_{FS}$	Chip Select Input Load Current		-0.05	-0.25	mA	$V_{CC}=5.5V$ , $V_S=0.45V$
$I_{RA}$	Address Input Leakage Current			40	$\mu A$	$V_{CC}=5.5V$ , $V_A = 5.5V$
$I_{RS}$	Chip Select Input Leakage Current			40	$\mu A$	$V_{CC}=5.5V$ , $V_S = 5.5V$
$V_{CA}$	Address Input Clamp Voltage		-0.9	-1.5	V	$V_{CC}=4.5V$ , $I_A=-10mA$
$V_{CS}$	Chip Select Input Clamp Voltage		-0.9	-1.5	V	$V_{CC}=4.5V$ , $I_S=-10mA$
$V_{OL}$	Output Low Voltage		0.3	0.45	V	$V_{CC}=4.5V$ , $I_{OL}=15mA$
$I_{CEX}$	3605A Output Leakage Current			40	$\mu A$	$V_{CC}=5.5V$ , $V_{CE}=5.5V$
$I_{CC}$	Power Supply Current		110	140	mA	$V_{CC}=5.5V$ , $V_{AO} \rightarrow V_{A9}=0V$ , $\overline{CS}_1=\overline{CS}_2=V_{IH}$
$V_{IL}$	Input "Low" Voltage			0.85	V	
$V_{IH}$	Input "High" Voltage	2.0			V	

### 3625, 3625-2 ONLY

Symbol	Parameter	Min.	Typ. <sup>[1]</sup>	Max.	Unit	Test Conditions
$I_{OI}$	Output Leakage for High Impedance Stage			40	$\mu A$	$V_O=5.5V$ or $0.45V$ , $V_{CC}=5.5V$ , $\overline{CS}_1=\overline{CS}_2=2.4V$
$I_{SC}^{[1]}$	Output Short Circuit Current	-20	-35	-80	mA	$V_O = 0V$
$V_{OH}$	Output High Voltage	2.4			V	$I_{OH} = -2.4mA$ , $V_{CC} = 4.5V$

NOTES: 1. Unmeasured outputs are open during this test.



## A. C. Characteristics $V_{CC} = +5V \pm 10\%$ , $T_A = 0^\circ C$ to $+75^\circ C$

Symbol	Parameter	Max. Limits		Unit	Conditions
		3605A-1 3625A-1	3605A 3625A		
$t_{A++}, t_{A--}$ $t_{A+-}, t_{A-+}$	Address to Output Delay	50	60	ns	$\overline{CS}_1 = \overline{CS}_2 = V_{IL}$ to select the PROM.
$t_{S++}$	Chip Select to Output Delay	30	30	ns	
$t_{S--}$	Chip Select to Output Delay	30	30	ns	

## Capacitance <sup>(1)</sup> $T_A = 25^\circ C$ , $f = 1$ MHz

SYMBOL	PARAMETER	LIMITS		UNIT	TEST CONDITIONS
		TYP.	MAX.		
$C_{INA}$	Address Input Capacitance	3	8	pF	$V_{CC} = 5V$ $V_{IN} = 2.5V$
$C_{INS}$	Chip-Select Input Capacitance	4	8	pF	$V_{CC} = 5V$ $V_{IN} = 2.5V$
$C_{OUT}$	Output Capacitance	5	10	pF	$V_{CC} = 5V$ $V_{OUT} = 2.5V$

NOTE 1: This parameter is only periodically sampled and is not 100% tested.

## Switching Characteristics

### Conditions of Test:

Input pulse amplitudes - 2.5V

Input pulse rise and fall times of

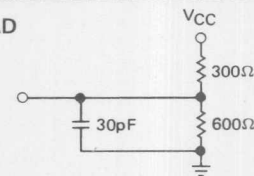
5 nanoseconds between 1 volt and 2 volts

Speed measurements are made at 1.5 volt levels

Output loading is 15 mA and 30 pF

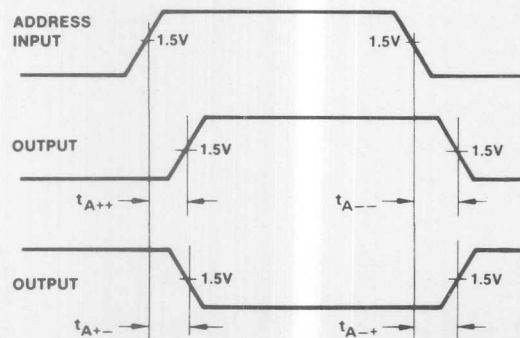
Frequency of test - 2.5 MHz

### 15mA TEST LOAD

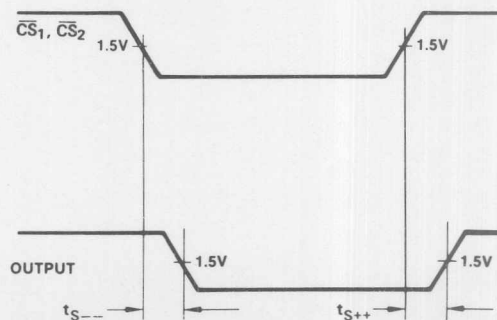


## Waveforms

### ADDRESS TO OUTPUT DELAY



### CHIP SELECT TO OUTPUT DELAY





# CHAPTER 6

## Development Aids

**MCS  
85**

**MCS  
85**

**MCS  
85**





## MODEL 220 INTELLEC® SERIES II MICROCOMPUTER DEVELOPMENT SYSTEM

**Complete Microcomputer Development System in one package for MCS-80, MCS-85 and MCS-48 microprocessor families**

**Integral CRT with detachable upper/lower case "typewriter" style full ASCII keyboard**

**Integral 250K-byte floppy disk with total storage capacity expandable to over 2M bytes**

**Single LSI electronics board with CPU, 32K bytes RAM memory and 4K bytes ROM memory**

**Built-in interfaces for High-Speed Paper Tape Reader/Punch, Printer and Universal PROM Programmer**

The Intellec Series II Model 220 is a complete microcomputer development system integrated into one compact package. It includes a CPU with 32K bytes of RAM memory, 4K bytes of ROM memory, a 2000-character CRT, detachable full ASCII keyboard with cursor controls and upper/lower case capability, and a 250K-byte floppy diskette drive.

Powerful ISIS-II Diskette Operating System software allows the Model 220 to be used quickly and efficiently for assembly and debugging of programs for Intel's MCS-80, MCS-85 or MCS-48 microprocessor families without the need for handling paper tape. ISIS-II performs all file handling operations for the user, leaving him free to concentrate on the details of his own application. When used in conjunction with an optional in-circuit emulator (ICE™) module, the Model 220 provides all the hardware and software development tools necessary for the rapid development of a microcomputer based product.

**Eight-level nested, maskable priority interrupt system**

**Powerful ISIS-II Diskette Operating System with Relocating Macro Assembler, Linker and Locator**

**Self-Test Diagnostic capability**

**Standard MULTIBUS with multiprocessor and DMA capability**

**Compatible with standard Intellec/iSBC Expansion Modules**

**Software compatible with previous Intellec Systems**



## MODEL 220 HARDWARE DESCRIPTION

The Intellec Series II Model 220 is a packaged, highly integrated microcomputer development system consisting of a CRT chassis with 6-slot cardcage, power supply, fans, cables, single floppy diskette drive and two printed circuit cards. A separate, full ASCII keyboard is connected with a cable. The master CPU card contains its own microprocessor, memory, I/O, interrupt and bus interface circuitry, fashioned from Intel's high-technology LSI components. Known as the integrated processor board (IPB), it occupies the first slot in the cardcage. A second, slave CPU card, is responsible for all remaining I/O control, including the CRT and keyboard interface and floppy disk control. This card, mounted on the rear panel, also contains its own microprocessor, RAM and ROM memory and I/O interface, thus in effect creating a dual processor environment. Known as the I/O controller (IOC), the slave CPU card communicates with the IPB over an 8-bit bidirectional data bus, thus leaving the remaining 5 slots in the cardcage available for system expansion.

The heart of the IPB is an Intel NMOS 8-bit microprocessor, the 8080A-2, running at 2.6 MHz. 32K bytes of RAM memory are provided on the board using Intel 16K RAMs. 4K of ROM is provided, preprogrammed with system bootstrap, "self-test" diagnostics and the Intellec Series II System Monitor. The 8-level vectored priority interrupt system allows interrupts to be individually masked. Using Intel's versatile 8259 interrupt controller, the interrupt system may be user-programmed to respond to individual needs.

The I/O subsystem in the Model 220 consists of two parts: the IOC card and two serial channels on the IPB itself. Each serial channel is RS232 compatible and is capable of running asynchronously from 110 to 9600 baud or synchronously from 150 to 56K baud. Both may be connected to a user-defined data set or data terminal. One channel contains current loop adapters. Both channels are implemented using Intel's 8251 USART. They can be programmatically selected to perform a variety of I/O functions. Baud rate selection is accomplished programmatically through an Intel 8253 Interval Timer. The 8253 also serves as a real-time clock for the entire system. I/O activity through both serial channels is signaled to the system through a second 8259 interrupt controller, operating in a polled mode, nested to the primary 8259.

The remainder of system I/O activity takes place in the IOC. The IOC provides interfaces for the CRT, keyboard, integral floppy disk and standard Intellec peripherals, including printer, high-speed paper tape reader/punch and Universal PROM Programmer. The IOC contains its own independent microprocessor, also an 8080A-2. This CPU controls all I/O operations, as well as supervising communications with the IPB. 8K bytes of ROM contain all I/O control firmware. 8K bytes of RAM are used for CRT screen refresh storage and the floppy disk buffer. These do not occupy any space in Intellec Series II main memory since the IOC is a totally independent microcomputer subsystem.

The CRT is a 12-inch raster scan-type monitor with a 50/60 Hz vertical scan rate and 15.5 kHz horizontal scan rate. Controls are provided for brightness and contrast adjustments. The interface to the CRT is provided through an Intel 8275 single-chip, programmable CRT controller. The master processor on the IPB transfers a character for display to the IOC, where it is stored in RAM. The CRT controller reads a line at a time into its line buffer through an Intel 8257 DMA Controller and then feeds one character at a time to the character generator to produce the video signal. Timing for the CRT control is provided by an Intel 8253 Interval Timer. The screen display is formatted as 25 rows of 80 characters. The full set of ASCII characters are displayed, including lower-case alphas.

The keyboard interfaces directly to the IOC processor via an 8-bit data bus. The keyboard contains an Intel UPI-41 Universal Peripheral Interface which scans the keyboard, encodes the characters and buffers the characters to provide N-key rollover. The keyboard itself is a high quality typewriter-style keyboard containing the full ASCII character set. An upper/lower case switch allows the system to be used for document preparation. Cursor control keys are also provided.

The floppy disk drive is controlled by an Intel 8271 single-chip, programmable floppy disk controller. It transfers data via an Intel 8257 DMA Controller between an IOC RAM buffer and the diskette. The 8271 handles reading and writing of data, formatting diskettes and reading status, all upon appropriate commands from the IOC microprocessor.

A UPI-41 Universal Peripheral Interface on the IOC board performs similar functions to the UPI-41 on the PIO board in the Model 210. It provides interface for other standard Intellec peripherals, including:

- Printer
- High-Speed Paper Tape Reader
- High-Speed Paper Tape Punch
- Universal PROM Programmer

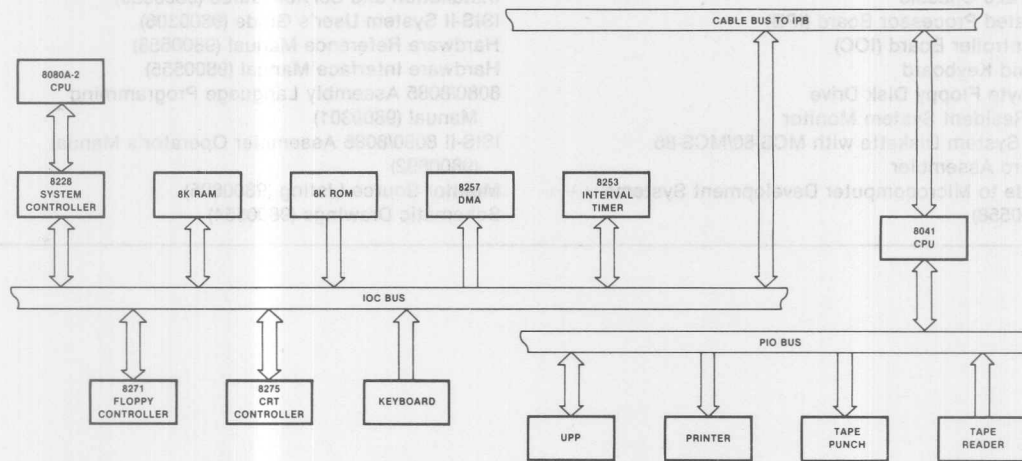
Communication between the IPB and IOC is maintained over a separate, 8-bit bidirectional data bus. Connectors for the devices named above, as well as the two serial channels, are mounted directly on the IOC itself.

User control is maintained through a front panel consisting of a power switch and indicator, reset/boot switch, run/halt light, and 8 interrupt switches and indicators. The front-panel circuit board is attached directly to the IPB, allowing the 8 interrupt switches to connect to the primary 8259, as well as the Intellec Series II Bus.

All Intellec Series II models implement the industry-standard MULTIBUS. It enables several bus masters, such as CPU and DMA devices, to share the bus and memory by operating at different priority levels. Resolution of bus exchanges is synchronized by a bus clock signal which is derived independently from processor clocks. Read/write transfers may take place at rates up to 5 MHz. The bus structure is suitable for use with any Intel microcomputer family.

The Model 220 may be expanded to 64K of RAM and up to 2¼ million bytes of on-line diskette storage.

## MODEL 220



### I/O CONTROLLER (IOC)

#### SPECIFICATIONS

##### PHYSICAL

Dimensions: 19.13" (48.59 cm) deep × 17.37" (44.12 cm) wide × 15.81" (40.16 cm) high

Weight: 86 lb (39 kg)

Keyboard: 9" (22 cm) deep × 17.37" (44.12 cm) wide × 3.0" (7.62 cm) high

Weight: 6 lb (3 kg)

##### ELECTRICAL

DC Power Supply:

Volts Supplied	Amps Supplied	Typical System Requirements
+ 5 ±5%	30	7.5
+12 ±5%	2.5	0.2
-12 ±5%	0.3	0.05
-10 ±5%	1.5	0.15
+15 ±5%	1.5	1.3*
+24 ±5%	1.7	1.2*

\*Not available on bus.

AC Requirements: 50-60 Hz, 115/230 VAC

##### ENVIRONMENTAL

Operating Temperature: 0° to 35°C (95°F)

##### HOST PROCESSOR (IPB)

8080A-2 based, operating at 2.600 MHz.

RAM: 32K, expandable to 64K with SBC-032 RAM boards (System Monitor occupies 62K through 64K).

ROM: 4K (2K in monitor, 2K in boot/diagnostic).

Bus: MULTIBUS, maximum transfer rate of 5 MHz.

Clocks: Host Processor, crystal controlled at 2.6 MHz.

Bus Clock, crystal controlled at 9.8304 MHz.

##### I/O Interfaces:

2 Serial I/O Channels, RS232C, at 110-9600 baud (asynchronous) or 150-56K baud (synchronous). Baud rates and serial format fully programmable using Intel 8251 USARTs. Serial Channel 1 additionally provided with 20 mA current loop.

Parallel I/O interfaces provided for paper tape punch, paper tape reader, printer, and Universal PROM Programmer.

##### Interrupts:

8-level, maskable, nested priority interrupt network initiated from front panel or user-selected devices.

##### Direct Memory Access (DMA):

Standard capability on MULTIBUS; implemented for user-selected DMA devices through optional DMA module — maximum transfer rate of 2 MHz.

##### Memory Access Time:

RAM: 585 ns

PROM: 450 ns

Diskette System Capacity: 250K bytes (Formatted)

##### Diskette Performance:

Diskette System Transfer Rate: 160K bits/sec.

##### Diskette System Access Time:

Track-to-Track: 10 ms

Average Random Positioning: 260 ms

Rotational Speed: 360 rpm

Average Rotational Latency: 83 ms

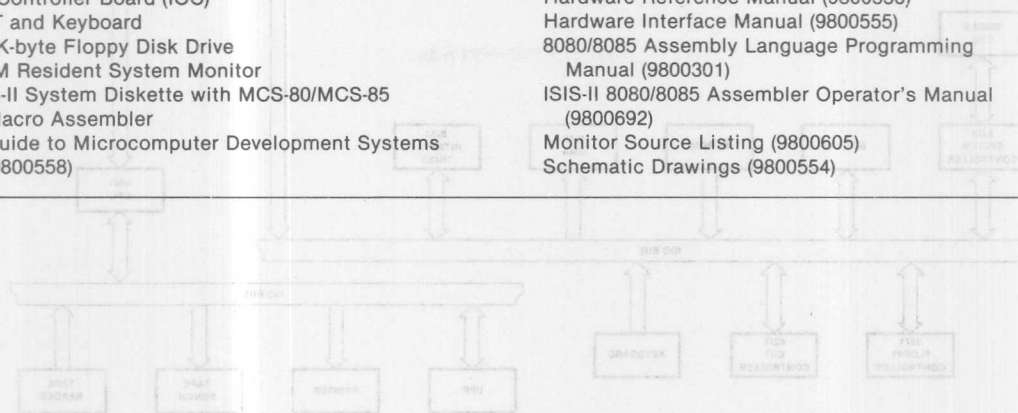
Recording Mode: FM

## MODEL 220

### EQUIPMENT SUPPLIED

Model 220 Chassis  
Integrated Processor Board (IPB)  
I/O Controller Board (IOC)  
CRT and Keyboard  
250K-byte Floppy Disk Drive  
ROM Resident System Monitor  
ISIS-II System Diskette with MCS-80/MCS-85  
Macro Assembler  
A Guide to Microcomputer Development Systems  
(9800558)

Installation and Service Guide (9800559)  
ISIS-II System User's Guide (9800306)  
Hardware Reference Manual (9800556)  
Hardware Interface Manual (9800555)  
8080/8085 Assembly Language Programming  
Manual (9800301)  
ISIS-II 8080/8085 Assembler Operator's Manual  
(9800692)  
Monitor Source Listing (9800605)  
Schematic Drawings (9800554)



IO CONTROLLER (IOC)

### SPECIFICATIONS

#### PHYSICAL

Dimensions: 18.13" (45.98 cm) deep x 17.31" (44.12 cm) wide x 15.81" (40.18 cm) high  
Weight: 8.16 (18 kg)  
Keyboard: 9" (22 cm) deep x 17.31" (44.12 cm) wide x 3.0" (7.62 cm) high  
Weight: 8.16 (18 kg)

#### ELECTRICAL

DC Power Supply:

Voltage	Amperes	Typical System Requirements
+5V ±5%	30	1.5
+12V ±5%	2.5	0.2
-10V ±5%	0.3	0.05
-10V ±5%	0.3	0.10
+15V ±5%	1.5	1.3*
+24V ±5%	1.7	1.5*

\*Not available in our

AC Requirements: 50-60 Hz, 115/230 VAC

#### ENVIRONMENTAL

Operating Temperature: 0° to 35°C (32°F)

#### HOST PROCESSOR (IPB)

8080A-2 based, operating at 2.00 MHz  
RAM: 32K, expandable to 64K with 88C-032 RAM boards (system monitor occupies 32K through 64K)

ROM: 4K (2K in memory, 2K in nonvolatile)  
Bus: MULTIBUS, maximum transfer rate of 5 MHz  
Clocks: Host processor crystal controlled at 2.0 MHz  
Bus Clock, crystal controlled at 8.3304 MHz  
IO interfaces:  
5 Serial IO Channels: RS232C at 110-9600 baud (asynchronous) or RS-485 (synchronous) baud rates and serial format fully programmable using Intel 8251 USART. Serial Channel 1 additionally provided with 20 mA current loop.  
Parallel IO interfaces provided for paper tape reader, paper tape reader, punch and Universal PROM Programmer.  
Interrupts:  
8-level, maskable, nested priority interrupt network initiated from front panel or user-selected device.  
Direct Memory Access (DMA):  
Standard capability on MULTIBUS, implemented for user-selected DMA devices through optional DMA module — maximum transfer rate of 5 MHz.  
Memory Access Time:  
RAM: 385 ns  
PROM: 450 ns  
Diskette System Capacity: 250K bytes (formatted)  
Diskette Performance:  
Diskette System Transfer Rate: 180K/byte/sec  
Diskette System Access Time:  
Track-to-Track: 10 ms  
Average Random Positioning: 280 ms  
Rotational Speed: 360 rpm  
Average Rotational Latency: 83 ms  
Recording Mode: F1



## MODEL 230 INTELLEC® SERIES II MICROCOMPUTER DEVELOPMENT SYSTEM

**Complete Microcomputer Development Center for Intel MCS-80, MCS-85 and MCS-48 microprocessor families**

**Integral CRT with detachable upper/lower case "typewriter-style" full ASCII keyboard**

**64K bytes RAM memory**

**1 million bytes (expandable to 2.5M bytes) of diskette storage**

**LSI electronics board with CPU, RAM, ROM, I/O and interrupt circuitry**

**Built-in interfaces for High-Speed Paper Tape Reader/Punch, Printer and Universal PROM Programmer**

The Intellec Series II Model 230 Microcomputer Development System is a complete center for the development of microcomputer-based products. It includes a CPU, 64K bytes of RAM, 4K bytes of ROM memory, a 2000-character CRT, detachable full ASCII keyboard and dual double-density diskette drives providing over 1 million bytes of on-line data storage.

Powerful ISIS-II Diskette Operating System software allows the Model 230 to be used quickly and efficiently for assembly and/or compilation and debugging of programs for Intel's MCS-80, MCS-85 or MCS-48 microprocessor families without the need for handling paper tape. ISIS-II performs all file handling operations for the user, leaving him free to concentrate on the details of his own application. When used in conjunction with an optional in-circuit emulator (ICE™) module, the Model 230 provides all the hardware and software development tools necessary for the rapid development of a microcomputer-based product.

**Powerful ISIS-II Diskette Operating System Software with Relocating Macro Assembler, Linker and Locater**

**"Self-Test" Diagnostic capability**

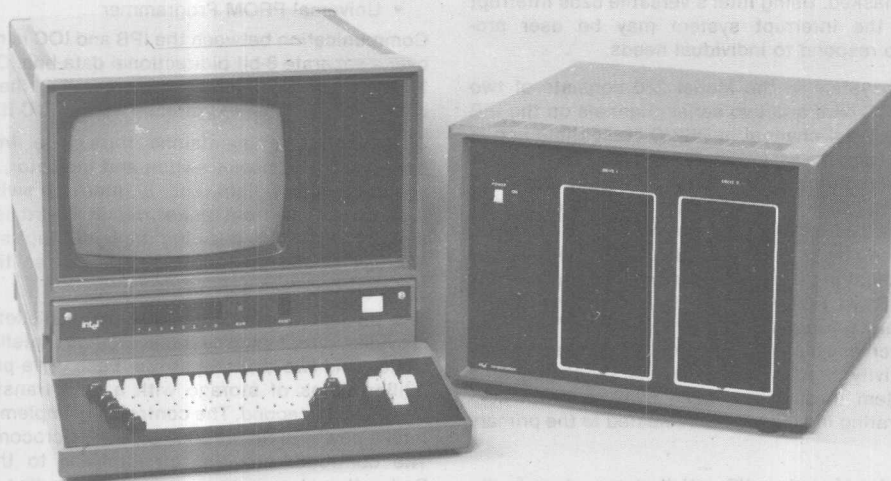
**Standard MULTIBUS with multiprocessor and DMA capability**

**Eight-level nested, maskable priority interrupt system**

**Compatible with standard Intellec/iSBC Expansion Modules**

**Software compatible with previous Intellec Systems**

**Supports PL/M and FORTRAN high level languages**



## MODEL 230 HARDWARE DESCRIPTION

The Intellec Series II Model 230 is a packaged, highly integrated microcomputer development system consisting of a CRT chassis with 6-slot cardcage, power supply, fans, cables, and five printed circuit cards. A separate, full ASCII keyboard is connected with a cable. A second chassis contains two floppy disk drives capable of double-density operation along with a separate power supply, fans and cables for connection to the main chassis.

The master CPU card contains its own microprocessor, memory, I/O, interrupt and bus interface circuitry fashioned from Intel's high technology LSI components. Known as the integrated processor board (IPB), it occupies the first slot in the cardcage. A second slave CPU card is responsible for all remaining I/O control including the CRT and keyboard interface. This card, mounted on the rear panel, also contains its own microprocessor, RAM and ROM memory, and I/O interface logic, thus, in effect, creating a dual processor environment. Known as the I/O controller (IOC), the slave CPU card communicates with the IPB over an 8-bit bidirectional data bus. In addition, 32K bytes of RAM (bringing the total to 64K bytes) is located on a separate card in the main cardcage. Fabricated from Intel's 16K RAMs, the board also contains all necessary address decoding and refresh logic. Two additional boards in the cardcage are used to control the two double-density floppy disk drives. Two remaining slots in the cardcage are available for system expansion. Additional expansion of 4 slots can be achieved through the addition of an Intellec Series II Expansion Chassis.

The heart of the IPB is an Intel NMOS 8-bit microprocessor, the 8080A-2, running at 2.6 MHz. 32K bytes of RAM memory are provided on the board using Intel 16K RAMs. 4K of ROM is provided, preprogrammed with system bootstrap, "self-test" diagnostics and the Intellec Series II System Monitor. The 8-level vectored priority interrupt system allows interrupts to be individually masked. Using Intel's versatile 8259 Interrupt Controller, the interrupt system may be user programmed to respond to individual needs.

The I/O subsystem in the Model 230 consists of two parts: the IOC card and two serial channels on the IPB itself. Each serial channel is RS232 compatible and is capable of running asynchronously from 110 to 9600 baud or synchronously from 150 to 56K baud. Both may be connected to a user-defined data set or terminal. One channel contains current loop adapters. Both channels are implemented using Intel's 8251 USART. They can be programmatically selected to perform a variety of I/O functions. Baud rate selection is accomplished programmatically through an Intel 8253 Interval Timer. The 8253 also serves as a real-time clock for the entire system. I/O activity through both serial channels is signaled to the system through a second 8259 interrupt controller, operating in a polled mode nested to the primary 8259.

The remainder of system I/O activity takes place in the IOC. The IOC provides interface for the CRT, keyboard, and standard Intellec peripherals including printer, high-

speed paper tape reader/punch and Universal PROM Programmer. The IOC contains its own independent microprocessor, also an 8080A-2. The CPU controls all I/O operations as well as supervising communications with the IPB. 8K bytes of ROM contain all I/O control firmware. 8K bytes of RAM are used for CRT screen refresh storage. These do not occupy space in Intellec Series II main memory since the IOC is a totally independent microcomputer subsystem.

The CRT is a 12-inch raster scan type monitor with a 50/60 Hz vertical scan rate and 15.5 kHz horizontal scan rate. Controls are provided for brightness and contrast adjustments. The interface to the CRT is provided through an Intel 8275 Single Chip Programmable CRT Controller. The master processor on the IPB transfers a character for display to the IOC, where it is stored in RAM. The CRT controller reads a line at a time into its line buffer through an Intel 8257 DMA Controller and then feeds one character at a time to the character generator to produce the video signal. Timing for the CRT control is provided by an Intel 8253 Interval Timer. The screen display is formatted as 25 rows of 80 characters. The full set of ASCII characters are displayed, including lower case alphas.

The keyboard interfaces directly to the IOC processor via an 8-bit data bus. The keyboard contains an Intel UPI-41 Universal Peripheral Interface which scans the keyboard, encodes the characters and buffers the characters to provide N-key rollover. The keyboard itself is a high quality typewriter style keyboard containing the full ASCII character set. An upper/lower case switch allows the system to be used for document preparation. Cursor control keys are also provided.

A UPI-41 Universal Peripheral Interface on the IOC board performs similar functions to the UPI-41 on the PIO board in the Model 210. It provides interface for other standard Intellec peripherals including:

- Printer
- High-Speed Paper Tape Reader
- High-Speed Paper Tape Punch
- Universal PROM Programmer

Communication between the IPB and IOC is maintained over a separate 8-bit bidirectional data bus. Connectors for the 4 devices named above, as well as the two serial channels, are mounted directly on the IOC itself.

User control is maintained through a front panel, consisting of a power switch and indicator, reset/boot switch, run/halt light and 8 interrupt switches and indicators. The front panel circuit board is attached directly to the IPB, allowing the 8 interrupt switches to connect to the primary 8259, as well as the Intellec Series II Bus.

The Intellec Series II double-density diskette system provides direct access bulk storage, intelligent controller, and two diskette drives. Each drive provides 1/2 million bytes of storage with a data transfer rate of 500,000 bits/second. The controller is implemented with Intel's powerful Series 3000 Bipolar Microcomputer Set. The controller provides an interface to the Intellec Series II system bus, as well as supporting up to four diskette drives. The diskette system records all data in soft sector format.

## MODEL 230

The diskette controller consists of two boards, the Channel Board and the Interface Board. These two PC boards reside in the Intellec Series II system chassis and constitute the diskette controller.

The Channel Board receives, decodes and responds to channel commands from the 8080A-2 CPU in the Model 230. The Interface Board provides the diskette controller with a means of communication with the diskette drives and with the Intellec system bus. The Interface Board validates data during reads using a cyclic redundancy check (CRC) polynomial and generates CRC data during write operations. When the diskette controller requires access to Intellec system memory, the Interface Board requests and maintains DMA master control of the system bus, and generates the appropriate memory command. The Interface Board also acknowledges I/O commands as required by the Intellec bus.

The diskette system is capable of performing seven different operations: recalibrate, seek, format track, write data, write deleted data, read data, and verify CRC.

In addition to supporting a second set of double-density drives, the diskette controller may co-reside with the Intel single density controller to allow up to 2.5 million bytes of on-line storage.

All Intellec Series II models implement the industry standard MULTIBUS. It enables several bus masters such as CPU and DMA devices to share the bus and memory by operating at different priority levels. Resolution of bus exchanges is synchronized by a bus clock signal which is derived independently from processor clocks. Read/write transfers may take place at rates up to 5 MHz. The bus structure is suitable for use with any Intel microcomputer family.

### SPECIFICATIONS

#### PHYSICAL

Dimensions:	19.13" (48.59 cm) deep × 17.37" (44.12 cm) wide × 15.81" (40.16 cm) high
Weight:	73 lb (33 Kg)
Keyboard:	9" (22.86 cm) deep × 17.37" (44.12 cm) wide × 3.0" (7.62 cm) high
Weight:	6 lb (3 Kg)
Dual Drive Chassis:	19.0" (48.26 cm) deep × 16.88" (42.88 cm) wide × 12.08" (30.68 cm) high
Weight:	64 lb (29 Kg)

#### ELECTRICAL

DC Power Supply:

Volts Supplied	Amps Supplied	Typical System Requirements
+5 ±5%	30	14.25
+12 ±5%	2.5	0.2
-12 ±5%	0.3	0.05
-10 ±5%	1.5	15
* +15 ±5%	1.5	1.3
* +24 ±5%	1.7	

\* Not available on bus.

AC Requirements: 50/60 Hz, 115/230 VAC

#### ENVIRONMENTAL

Operating Temperature: 0° to 35°C (95°F)

#### HOST PROCESSOR (IPB)

RAM: 64K (System Monitor occupies 62K through 64K).  
ROM: 4K (2K in monitor, 2K in boot/diagnostic).

#### Diskette System Capacity (Basic Two Drives):

Unformatted
Per Disk: 6.2 megabits
Per Track: 82.0 kilobits
Formatted
Per Disk: 4.1 megabits
Per Track: 53.2 kilobits

#### Diskette Performance:

Diskette System Transfer Rate: 500 kilobits/sec
Diskette System Access Time
Track-to-Track: 10 ms
Head Settling Time: 10 ms
Average Random Positioning Time: 260 ms
Rotational Speed: 360 rpm
Average Rotational Latency: 83 ms
Recording Mode: M <sup>2</sup> FM

#### EQUIPMENT SUPPLIED

Model 230 Chassis  
Integrated Processor Board (IPB)  
I/O Controller Board (IOC)  
32K RAM Board  
CRT and Keyboard  
Double-Density Floppy Disk Controller (2 boards)  
Dual-Drive Floppy Disk Chassis and Cables  
2 Floppy Disk Drives (512K byte capacity each)  
ROM-Resident System Monitor  
ISIS-II System Diskette with MCS-80/MCS-85 Macro Assembler  
A Guide to Microcomputer Development Systems (9800558)  
Installation and Service Guide (9800550)  
ISIS-II System User's Guide (9800306)  
Hardware Reference Manual (9800556)  
Hardware Interface Manual (9800555)  
8080/8085 Assembly Language Programming Manual (9800301)  
ISIS-II 8080/8085 Assembler Operator's Manual (9800292)  
Monitor Source Listing (9800605)  
Schematic Drawings (9800554)





## ISIS-II

# DISKETTE OPERATING SYSTEM MICROCOMPUTER DEVELOPMENT SYSTEM

Supports up to four double density drives and two single density drives, providing up to 2.5 Mega-bytes of storage in one system with up to 200 files per diskette

Supports resident, high level programming languages, PL/M and FORTRAN

Relocating MCS-80/MCS-85 macro assembler contains extended macro and conditional assembly capability

Linker automatically combines separately assembled or compiled programs into a single relocatable module

Library Manager™ creates and updates program libraries

Command file facility allows console commands to be submitted from a diskette file

Diskette system text editor provides string search, substitution, insertions, and deletion commands

Diskette operating system functions are callable from user programs

Access to all Intellec monitor facilities provided

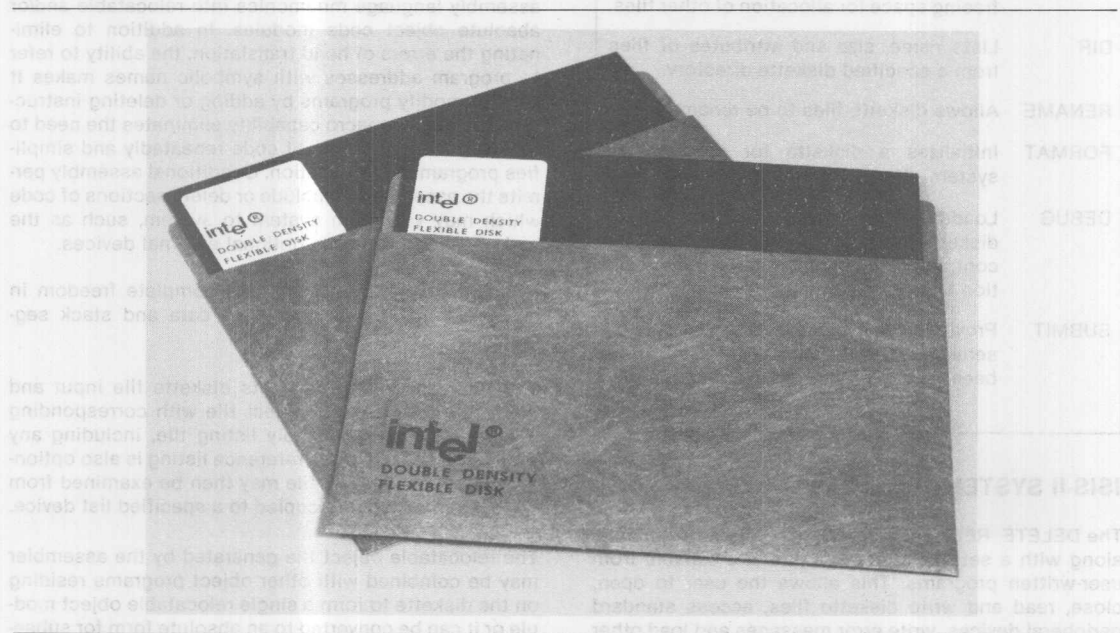
Dynamic allocation and de-allocation of diskette sectors for variable length files

Supports all standard Intellec peripherals

The ISIS-II Diskette Operating System is a sophisticated, general purpose, high-speed data handler and file manipulation system. It provides the ability to edit, assemble, compile, link, relocate, execute and debug programs, and performs all file management tasks for the user.

The ISIS-II operating system resides on the system diskette and supports a broad range of user-oriented design aid software. Total file management and input editing features greatly reduce software development time. The ISIS-II Relocating Macro Assembler, Linker, Object Locator and Library Manager can be loaded from the diskette in seconds. All passes of the assembler can be executed without the need for user intervention. Object code and listings may be directed to any output device, or stored as diskette files.

Powerful system console commands are provided in an easy-to-use context. Monitor mode can be entered by a special prefix to any system command or program call.



## ISIS-II FILES

A file is a user-defined collection of information of variable length. ISIS-II also treats each of the standard Intellec® system peripherals as files through pre-assignment of unique file names to each device. In this manner data can be copied from one device to another (i.e., tape reader to tape punch) using the same command required to copy one diskette data file to another. ISIS-II provides automatic implementation of random access disk files. Each file is identified by a user-chosen name unique on its diskette. Up to 200 files may be stored on each diskette.

## ISIS-II SYSTEM COMMANDS

ISIS-II system commands are designed to provide the user with a powerful, easy-to-use program and file manipulation capability. Several commands have the capability of operating on several files at once via the wildcard file-naming convention. As an example, the command "DELETE \*.OBJ" deletes all files in the diskette directory with the suffix ".OBJ".

DISK	Initializes a diskette for use by the system. Requires only one disk drive.
ATTRIB	Assigns specified attributes to a file, such as write-protect.
COPY	Creates copies of existing diskette files or transfers files from one device to another.
DELETE	Removes a file from the diskette, thereby freeing space for allocation of other files.
DIR	Lists name, size and attributes of files from a specified diskette directory.
RENAME	Allows diskette files to be renamed.
FORMAT	Initializes a diskette for use by the system. (Use with two or more drives.)
DEBUG	Loads a specified program from a diskette into memory and then transfers control to the Intellec monitor for execution and or debugging.
SUBMIT	Provides the capability to execute a series of ISIS-II commands which have been previously written to a diskette file.

## ISIS-II SYSTEM CALL CAPABILITY

The DELETE, RENAME and ATTRIB system commands, along with a set of file I/O routines, are callable from user-written programs. This allows the user to open, close, read and write diskette files, access standard peripheral devices, write error messages and load other programs via simple program call statements.

## ISIS-II TEXT EDITOR

The ISIS-II Text Editor is a comprehensive tool for the entry and correction of assembly language, PL/M and FORTRAN programs for Intel® microcomputers. Its command set allows manipulation of either entire lines of text or individual characters within a line.

Programs may be entered from the console keyboard or may be loaded directly. Text is stored internally in the editor's workspace, and may be edited with the following commands:

- string insertion or deletion
- string search
- string substitution

To facilitate the use of these editing commands, utility commands are used to change positions in the workspace. These include:

- move pointer by line or by character
- move pointer to start of workspace
- move pointer to end of workspace

The contents of the workspace are stored on diskette and can be immediately accessed by ISIS-II commands or other programs, such as the ISIS-II MCS-80/MCS-85 Macro Assembler.

## ISIS-II MCS-80/MCS-85 RELOCATING MACRO ASSEMBLER

The ISIS-II MCS-80/MCS-85 Macro Assembler translates assembly language mnemonics into relocatable and/or absolute object code modules. In addition to eliminating the errors of hand translation, the ability to refer to program addresses with symbolic names makes it easy to modify programs by adding or deleting instructions. Extended macro capability eliminates the need to rewrite similar sections of code repeatedly and simplifies program documentation. Conditional assembly permits the assembler to include or delete sections of code which may vary from system to system, such as the code required to handle optional external devices.

In addition, the user is allowed complete freedom in assigning the location of code, data and stack segments.

The ISIS-II Assembler accepts diskette file input and produces a relocatable object file with corresponding symbol table and assembly listing file, including any error messages. A cross reference listing is also optionally produced. The list file may then be examined from the system console or copied to a specified list device.

The relocatable object file generated by the assembler may be combined with other object programs residing on the diskette to form a single relocatable object module or it can be converted to an absolute form for subsequent loading and execution.

## ISIS-II DISKETTE OPERATING SYSTEM

### ISIS-II LINKER

The ISIS-II LINKER provides the capability to combine the outputs of several independently compiled or assembled object modules (files) into a single relocatable object module. The LINKER automatically resolves all external program and data references during the linking process.

Object modules produced from previous link operations may be easily linked to a new module. ISIS-II also provides facilities to ease the generation of overlays.

An optional link map showing the contents and lengths of each segment in the output module can be requested. All unsatisfied external references are also listed.

If requested by the user, the ISIS-II LINKER can search a specified set of program libraries for routines to be included in the output module.

### ISIS-II OBJECT LOCATOR

The ISIS-II LOCATE program takes output from either the resident FORTRAN or PL/M compilers, the macro

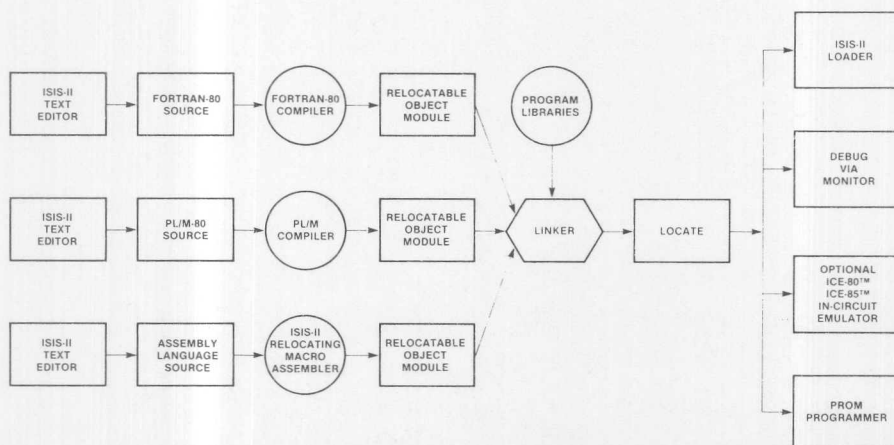
assembler or the LINKER and transforms that output from a relocatable format to an absolute format which may then be loaded via the standard ISIS-II loader, or loaded into the appropriate In-Circuit Emulator (ICE) module.

During the LOCATE process, code, data and stack segments can be *separately* relocated, allowing code to be put in areas to be subsequently specified as ROM, while data and the stack can be directed to RAM addresses.

A LOCATE map showing absolute addresses for each code and data segment and a symbol table dump listing symbols, attributes and absolute address can also be requested.

### ISIS-II LIBRARY MANAGER

The ISIS-II LIBRARY MANAGER program provides for the creation and maintenance of a program library containing Intel-provided and user-written programs and subroutines. These library routines can be linked to a program using the ISIS-II LINKER. Several libraries, each containing its own set of routines, can be created.



PROGRAM DEVELOPMENT FLOW USING ISIS-II DISK OPERATING SYSTEM





PL/M-80

## PL/M-80

# HIGH LEVEL PROGRAMMING LANGUAGE INTELLEC® RESIDENT COMPILER

**Cuts software development and maintenance costs**

**Speeds project completion**

**Improves product reliability**

**Eases enhancements as system capabilities expand**

**Produces relocatable and linkable object code**

**Resident operation on Intellec® Microcomputer Development System and Intellec® Series II Microcomputer Development Systems**

**Sophisticated code optimization reduces application memory requirements**

PL/M-80 is an advanced, high-level programming language for Intel® 8080 and 8085 Microprocessors, ISBC-80 OEM Computer Systems and Intellec® Microcomputer Development Systems. PL/M has been substantially enhanced since its introduction in 1973 and has become one of the most effective and powerful microprocessor systems implementation tools available. It is easy to learn, facilitates rapid program development and debugging, and significantly reduces maintenance costs.

PL/M is a powerful, high-level algorithmic language in which program statements can naturally express the algorithm to be programmed. This frees programmers to concentrate on their system development without having to deal with assembly language details (such as register allocation, meanings of assembler mnemonics, etc.).

The PL/M compiler efficiently converts free-form PL/M programs into equivalent 8080/8085 instructions. Substantially fewer PL/M statements are necessary for a given application than if it were programmed at the assembly language or machine code level.

Since PL/M programs are problem oriented and more compact, programming in PL/M results in a high degree of productivity during development efforts. This translates into significant reductions in software development and maintenance costs for the user.



## FEATURES

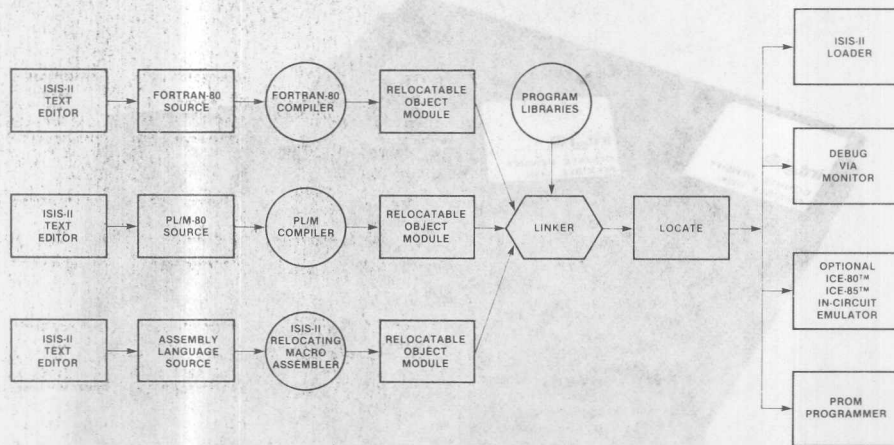
Major features of the Intel PL/M-80 Compiler and programming language include:

- Resident operation on the Intel® Microcomputer Development System eliminates the need for a large in-house computer or costly timesharing system.
- Generation of relocatable and linkable object code permits PL/M programs to be developed and debugged in small modules. These modules can be easily linked with other modules and/or library routines to form a complete application.
- Extensive code optimization results in generation of short, efficient CPU instruction sequences. Major optimizations include compile time arithmetic, constant subscript resolution, and common subexpression elimination.
- The PL/M Compiler fully supports symbolic debugging with the ICE-80™ and ICE-85™ In-Circuit Emulators.
- Compile time options include general listing format commands, symbol table listing, cross reference listing, and "innerlist" of generated assembly language instructions.
- Block structure aids in utilization of structured programming techniques.
- High level PL/M statements provide access to hardware resources (interrupt systems, absolute addresses, CPU input/output ports).
- Complex data structures may be defined at a high level.
- Re-entrant procedures may be specified as a user option.

## BENEFITS

PL/M is designed to be an efficient, cost-effective solution to the special requirements of microcomputer software development as illustrated by the following benefits of PL/M use:

- **Low Learning effort** — PL/M is very easy to learn even for the novice programmer.
- **Earlier Project Completion** — Critical projects are completed much earlier than otherwise possible because PL/M substantially increases programmer productivity.
- **Lower Development Cost** — Increases in programmer productivity translate into lower software development costs because less programming resources are required for a given function.
- **Increased Reliability** — PL/M is designed to assist in the development of reliable software (PL/M programs are simple statements of the program algorithm). This substantially reduces the risk of costly correction of errors in systems that have already reached full production status because a simply stated program is more likely to correctly perform its intended function.
- **Easier Enhancements and Maintenance** — Programs written in PL/M are easier to read and easier to understand. This means it is easier to enhance and maintain PL/M programs as system capabilities expand and future products are developed.
- **Simpler Project Development** — The Intel® Microcomputer Development System, with resident PL/M-80, is all that is needed for development and debugging of software for 8080 and 8085 microcomputers. This reduces development time and cost because expensive (and remote) timesharing or large computers are not required.



The PL/M Compiler is an efficient multiphase compiler that accepts source programs, translates them into object code, and produces requested listings. After compilation, the object program may be linked to other modules, located to a specific area of memory, then executed. The diagram shown above illustrates a program development cycle where the program consists of three modules, one PL/M, one Fortran, and the other assembly language.

PL/M-80	COMPILER	FACTORIAL GENERATOR – PROCEDURE
		\$OBJECT(:F1:FACT.OB2) \$DEBUG \$XREF \$TITLE('FACTORIAL GENERATOR – PROCEDURE') \$PAGEWIDTH(80)
1		FACT: DO;
2	1	DECLARE NUMCH BYTE PUBLIC;
3	1	FACTORIAL: PROCEDURE (NUM,PTR) PUBLIC;
4	2	DECLARE NUM BYTE, PTR ADDRESS;
5	2	DECLARE DIGITS BASED PTR (161) BYTE;
6	2	DECLARE (I,C,M) BYTE;
7	2	NUMCH=1; DIGITS(1)=1;
9	2	DO M = 1 TO NUM;
10	3	C=0;
11	3	DO I = 1 TO NUMCH;
12	4	DIGITS(I) = DIGITS(I) * M + C;
13	4	C = DIGITS(I)/10;
14	4	DIGITS(I) = DIGITS(I) – 10 * C;
15	4	END;
16	3	IF C <> 0 THEN
17	3	DO;
18	4	NUMCH = NUMCH+1; DIGITS (NUMCH) = C;
20	4	C = DIGITS(NUMCH)/10;
21	4	DIGITS(NUMCH) = DIGITS(NUMCH) – 10 * C;
22	4	END
		END;
24	2	END FACTORIAL;
25	1	END;

## SPECIFICATIONS

Required software  
ISIS-II Diskette Operating System

### Operating Environment:

#### Required hardware

Intellec® Microcomputer Development System  
65K bytes of memory  
Dual diskette drives  
System console — teletype

#### Documentation Package:

PL/M Programming Manual  
ISIS-II PL/M-80 Compiler Operator's Manual

#### Optional hardware

CRT as system console  
Line printer

Shipping Media:  
Diskette





## MICROCOMPUTER DEVELOPMENT SYSTEMS

### ICE-85™ MCS-85™ IN-CIRCUIT EMULATOR

Connects the Inteltec® System Resources to the user-configured system via a 40-pin adaptor plug

Executes user system software in real-time

Allows user-configured system to share Inteltec® memory and I/O facilities

Provides 1023 states of 8085 trace data plus 18 additional logic signals via an External Trace Module

Offers full symbolic debugging capability for both assembly language and Intel's high-level compiler language, PL/M-80

Displays trace data from the user's 8085 in assembler mnemonics and allows personality groupings of data sampled by the external 18-channel trace module

Extends ICE capabilities to the rest of the prototype system peripheral circuitry by allowing the user to execute his own peripheral chip analysis routines

Provides ability to examine and alter MCS-85™ registers, memory, flag values, interrupt bits and I/O ports

The ICE-85 module resides in the Inteltec® Microcomputer Development System and interfaces to the user system's 8085. In addition, an external trace module provides access to user system peripheral circuitry via a user-configured DIP clip for peripheral ICs or may be attached to as many as 18 separate prototype signal nodes via individual probe clips. Using the ICE-85 module, the designer can execute prototype software in real-time or single-step mode and can substitute Inteltec® system memory and I/O for user system equivalent. ICE capability can be extended to the rest of the user system peripheral circuitry by allowing the user to create and execute a library of user-defined peripheral chip analyzer routines. All user access to the prototype system software may be done symbolically by assigning names to program locations and data, I/O ports and groups of external trace signals. For the first time, in-circuit emulation extends beyond the user's prototype CPU to the entire user's system, allowing In-System Emulation.



## SYMBOLIC DEBUGGING CAPABILITY

ICE-85 allows the user to make symbolic references to I/O ports, memory addresses and data in his program. Symbols and PL/M statement number may be substituted for numeric values in any of the ICE-85 commands. The user is relieved from looking up addresses of variables or program subroutines.

The user symbol table generated along with the object file during a PL/M-80 compilation or by the ISIS-II 8080/8085 Macro Assembler is loaded into the Intellec® System memory along with the user program which is to be emulated. The user may add to this symbol table any additional symbolic values for memory addresses, constants, or variables that are found useful during system debugging. By referring to symbol memory addresses, the user can examine, change or break at the intended location.

ICE-85 provides symbolic definition of all 8085 registers, interrupt bits and flags. The following symbolic references are also provided for user convenience: TIMER, the low-order 16 bits of a register containing the number of 2 MHz clock pulses elapsed during emulation; HTIMER, the high-order 16 bits of the timer counter; PPC, the address of the last instruction emulated; BUFFERSIZE, the number of frames of valid trace data (between 0 and 1022).

## PERSONALITY GROUPED DISPLAYS

Trace data in the 1023 by 42-channel real-time trace memory buffer is displayed in easy to read format. The user has the option to specify trace data displays in actual 8085 assembler instruction mnemonics. The data collected from the External Trace Module can be grouped and symbolically named according to user specifications and displayed in the appropriate number base designation. Simple ICE-85 commands allow the user to select any portion of the 42K-bit trace buffer for immediate display.

## MEMORY AND I/O MAPPING

Memory and I/O for the user system can be resident in the user system or "borrowed" from the Intellec® System through ICE-85's mapping capability.

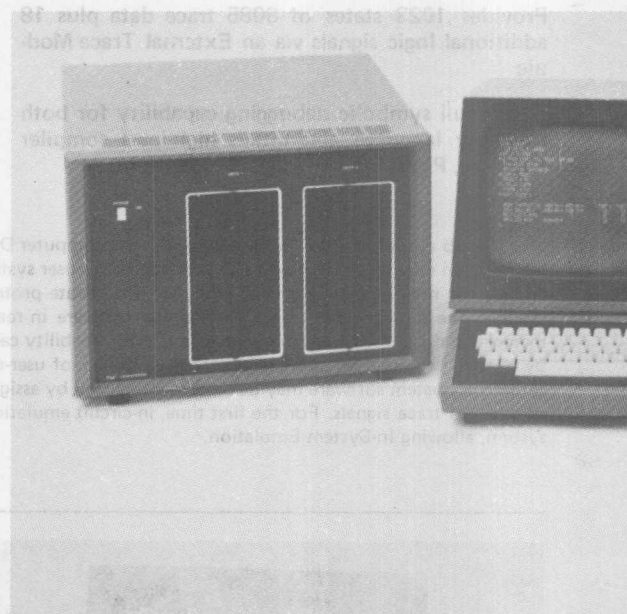
ICE-85 separates user memory into 32 2K blocks. Each block of memory can be defined independently. The user may assign Intellec® System equivalents to take the place of devices not yet designed for the user system during prototyping. In addition, Intellec® System memory or I/O can be accessed in place of suspect user system devices during prototyping or production checkout.

The user can also designate a block of memory or I/O as nonexistent. ICE-85 issues error messages when memory or I/O designated as nonexistent is accessed by the user program.

## INTEGRATED HARDWARE/SOFTWARE DEVELOPMENT

The user prototype need consist of no more than an 8085 CPU socket and a user bus to begin integration of software and hardware development efforts. Through ICE-85 mapping capabilities, Intellec® System equivalents can be accessed for missing prototype hardware. Hardware designs can be tested using the system software which will drive the final product.

The system integration phase, which can be so costly when attempting to mesh completed hardware and software products, becomes a convenient two-way debug tool when begun early in the design cycle.



## TYPICAL ICE INTERROGATION AND UTILITY COMMANDS

DISPLAY/CHANGE	Display/Changes the values of symbols and the contents of 8085 registers, pseudo-registers, status flags, interrupt bits, I/O ports and memory.
EVALUATE	Displays the value of an expression in the binary, octal, decimal or hexadecimal.
SEARCH	Searches user memory between locations in a user program for specified contents.
CALL	Emulates a procedure starting at a specified memory address in user memory.
ICALL	Executes a user-supplied procedure starting at a specified memory address in the Intellec® System memory.
EXECUTE	Saves emulated program registers and emulates a user-supplied subroutine to access peripheral chips in the user's system.

## REAL TIME TRACE

ICE-85 captures valuable trace information from the emulating CPU and the External Trace Module while the user is executing programs in real time. The 8085 status, the user memory or port addressed, the data read or written, the serial data lines and data from 18 external signals, is stored for the last 1023 machine states executed (511 machine cycles). This provides ample data for determining how the user system was reacting prior to emulation break. It is available whether the break was user-initiated or the result of an error condition.

For detailed information on the actions of CPU registers, flags, or other system operations, the user may operate in single or multi-step sequences tailored to system debug needs.

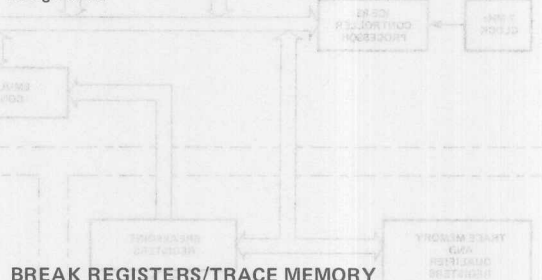


## EXTERNAL TRACE MODULE

TTL level signals from 18 points in the user system may be synchronously sampled by the External Trace Module and collected in ICE-85's trace buffer. The signals can be collected from a single peripheral chip via the supplied 40-pin DIP clip or may be placed by the user on up to 18 separate signal nodes using the supplied 18 individual probe clips. These signals are included in the 42-channel breakpoint comparisons and clock qualifiers. Also, data from these 18 channels may be displayed in each to read, user-defined groupings.

## SYNCHRONOUS OPERATION WITH OTHER DESIGN AIDS

ICE-85 can be synchronized with other Intellec® design aids by means of two external synchronization lines. These lines are used to enable and disable ICE-85 trace data collection and to cause break conditions based on an external signal which may not be included in the ICE-85 breakpoint registers. In addition, ICE-85 can generate signals on these lines which may be used to control other design aids.



## BREAK REGISTERS/TRACE MEMORY

ICE-85 has two breakpoint registers which are used to break emulation, and two trace qualifier registers which are used to control the collection of trace data during emulation. Each register is 42 entries wide, one entry for each channel and each entry can take any one of the three values 0, 1 or "don't care".

The trace buffer, also 42 entries wide, collects data sampled from 24 8085 processor channels and 18 external channels sampled by the External Trace Module. The signals collected from the 8085 include address lines, data lines, status lines and serial input and output lines. The 18 channels extending from the External Trace Module synchronously sample and collect into the trace buffer any user-specified TTL compatible signal from the rest of the prototype system. "Break" and "trace qualification" may therefore occur as a result of a match of any combination of up to 42 channels of CPU and external circuitry signals.

## EMULATION CONTROLS AND COMMANDS

GROUP	Defines into a symbolically named group, a channel or combination of channels from the 8085 Microprocessor and/or the External Trace Module.
GO	Initiates real-time emulation and controls emulation break conditions.
STEP	Initiates emulation in single instruction steps. User may specify the type and amount of information displayed following each step, and define conditions under which stepping should continue.
PRINT	Prints the user-specified portion of the trace memory to the selected list device.

# MCS-85™ IN-CIRCUIT EMULATOR

## SPECIFICATIONS

### ICE-85 OPERATING ENVIRONMENT

Diskette-Based ICE-85 Software

Required Hardware:

- Intellec® Microcomputer Development System
- System Console
- Intellec® Diskette Operating System
- ICE-85 Module

Required Software:

- System Monitor
- ISIS-II

### EQUIPMENT SUPPLIED

18-Channel External Trace Module

Printed Circuit Boards (2)

Interface Cable and Emulation Buffer Module

Operator's Manual

ICE-85 Software, Diskette-Based Version

### EMULATION CLOCK

User's system clock or ICE-85 adaptor socket  
(6.144 MHz Crystal)

### PHYSICAL CHARACTERISTICS

Printed Circuit Boards:

- Width: 12.00 in. (30.48 cm)
- Height: 6.75 in. (17.15 cm)
- Depth: 0.50 in. (1.27 cm)
- Packaged Weight: 6.00 lb (2.73 kg)

### ELECTRICAL CHARACTERISTICS

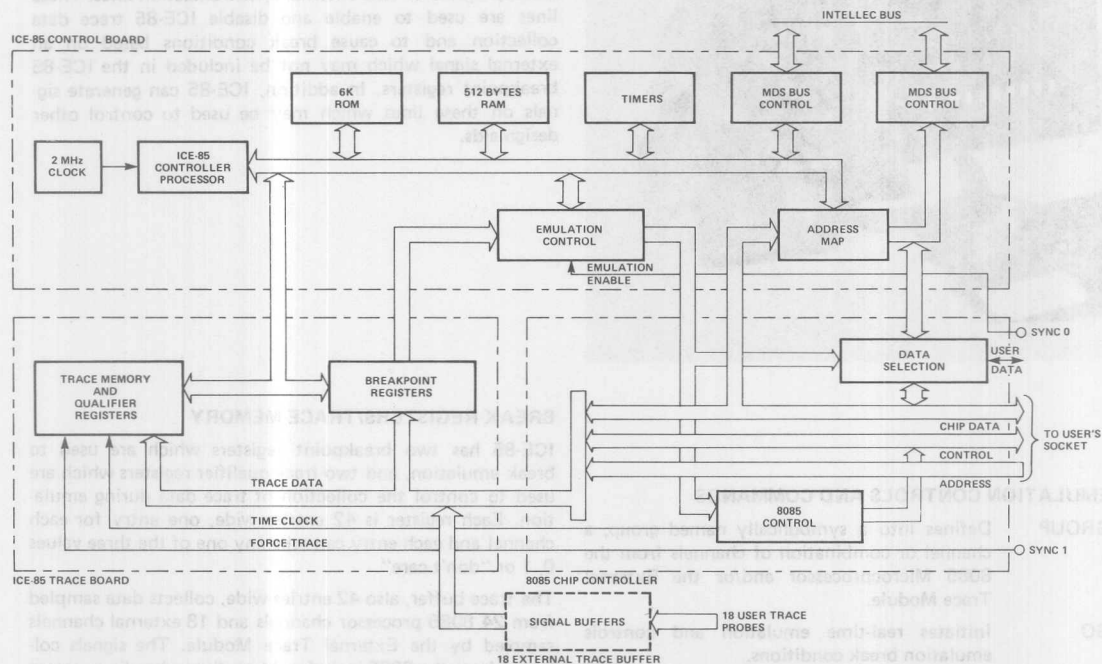
DC Power:

- $V_{CC} = +5V \pm 5\%$
- $I_{CC} = 12A$  maximum; 10A typical
- $V_{DD} = +12V \pm 5\%$
- $I_{DD} = 80$  mA maximum; 60 mA typical
- $V_{BB} = -10V \pm 5\%$
- $I_{BB} = 30$  mA maximum; 10  $\mu A$  typical

### ENVIRONMENTAL CHARACTERISTICS

Operating Temperature: 0° to 40°C

Operating Humidity: Up to 95% relative humidity without condensation.



ICE-85 BLOCK DIAGRAM

## ORDERING INFORMATION

Part Number	Description
MDS-85-ICE	8085 CPU In-Circuit Emulator and 18-Channel External Trace Module

## SDK-85

### MCS-85™ SYSTEM DESIGN KIT

**Complete Single Board Microcomputer System Including CPU, Memory and I/O**  
**Easy to Assemble Kit-Form**

**High-Performance 3MHz 8085 CPU**  
**(1.3  $\mu$ s Instruction Cycle)**

**Popular 8080A Instruction Set**  
**Interfaces Directly With TTY**

**Interactive LED Display and Keyboard**  
**Large Wire-Wrap area for Custom Interfaces**

**Extensive System Monitor Software in ROM**

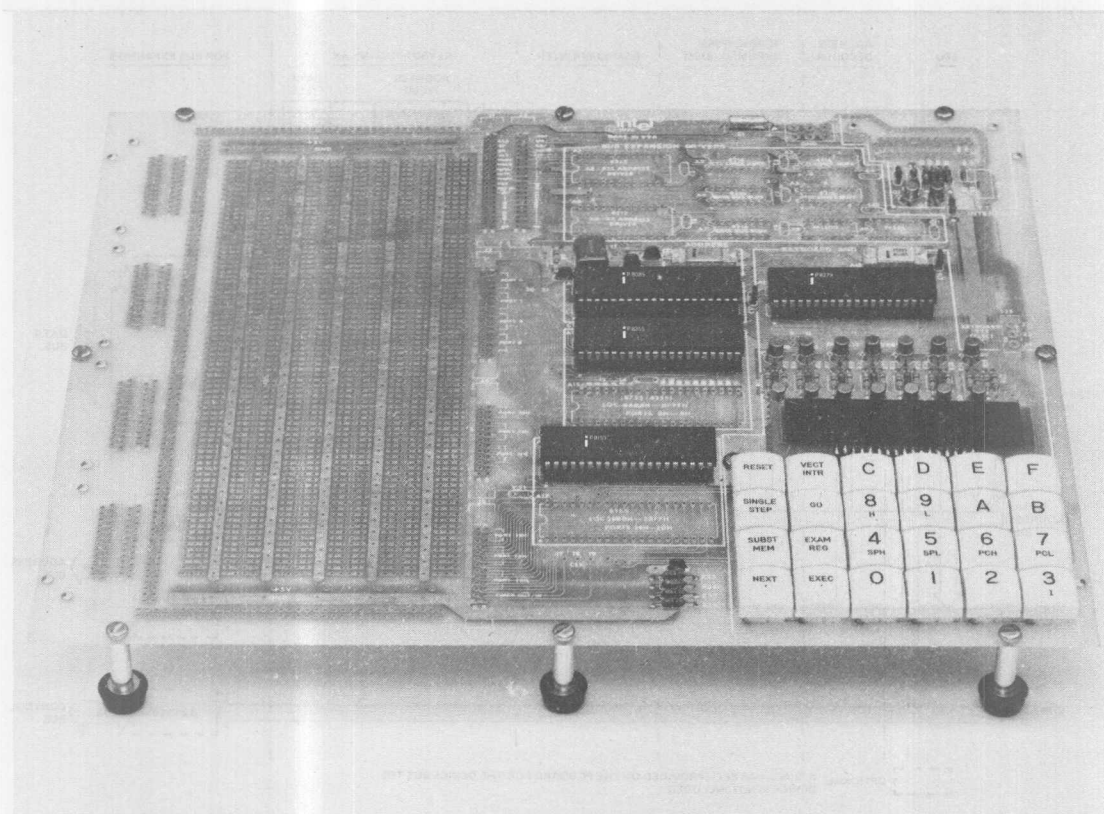
**Comprehensive Design Library Included**

**Low Cost**

The MCS-85 System Design Kit (SDK-85) is a complete, single board, microcomputer system in kit form. It contains all necessary components, including LED Display, Keyboard, resistors, caps, crystal and miscellaneous hardware to complete construction. Included is a preprogrammed ROM that contains the system monitor for general software utilities and system diagnostics.

The SDK-85 includes 6 digit LED display and 24 key-keyboard for a direct insertion, examination and execution of a user's program. In addition, it can be directly interfaced with a teletype terminal.

The SDK-85 is an inexpensive, high-performance prototype system that has designed-in flexibility for simple interface to the user's application.



The SDK-85 is a complete 8085 microcomputer system on a single board, in kit form. It contains all necessary components to build a useful, functional system. Such items as resistors, caps, and sockets are included. Assembly time varies from 3 to 5 hours, depending on the skill of the user.

A compact but powerful system monitor is supplied with the SDK-85 to provide general software utilities and system diagnostics. It comes in a pre-programmed ROM.

The SDK-85 communicates with the outside world through either the on-board LED Display/Keyboard combination or, the user's TTY terminal (Jumper Selectable). Both memory and I/O can be easily expanded by simply soldering in additional devices in locations provided for this purpose. A large area of the board (45 sq. in.) is laid out as general purpose wire-wrap for the user's custom interfaces.

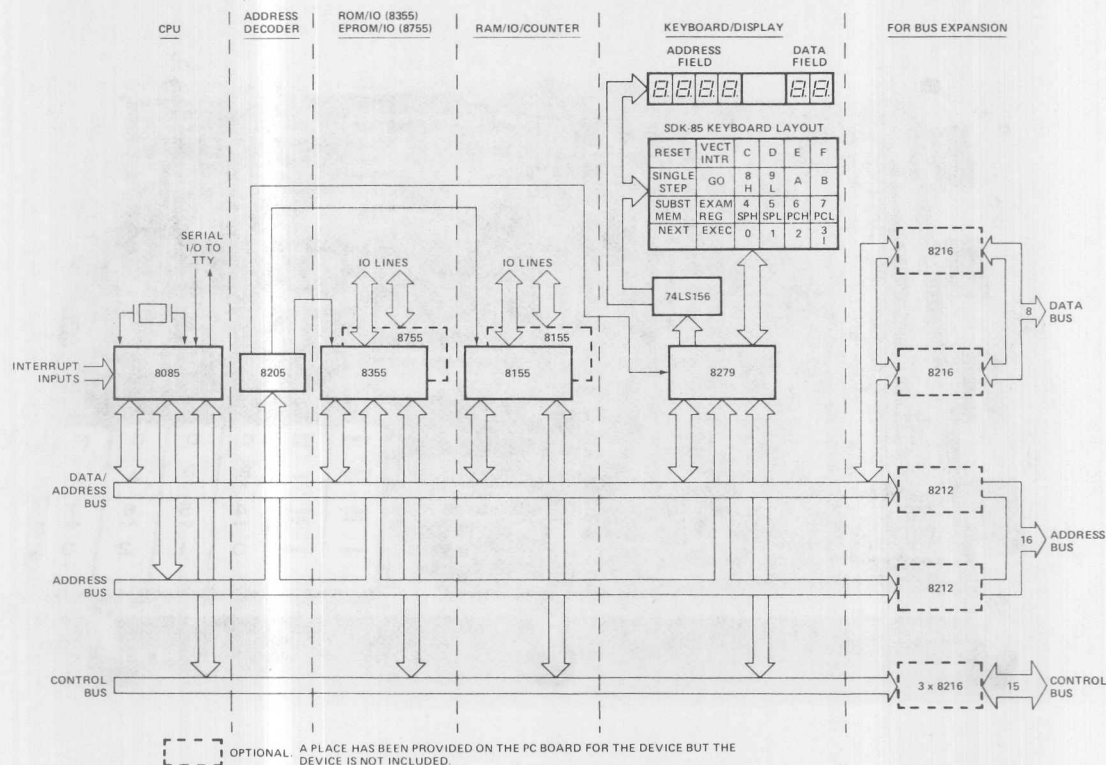
Only a few simple tools are required for assembly; soldering iron, cutters, screwdriver, etc. The SDK-85 User's Manual contains step-by-step instructions that make assembly easy, and eliminate mistakes. Once construction is complete, the user connects his kit to a power supply and the SDK-85 is ready to go. The monitor starts immediately upon power-on or reset.

- Reset — Starts the monitor
- GO — allows you to execute a user program
- Single Step — allows you to execute a user program one instruction at a time — useful for debugging
- Substitute Memory — allows you to examine and modify memory locations
- Examine Register — allows you to examine and modify the 8085's register contents
- Vector Interrupt — a user interrupt button

#### Teletype Monitor Commands

- Display Memory — displays multiple memory locations
- Substitute Memory — allows you to examine and modify memory locations one at a time
- Insert Instructions — allows you to store multiple bytes in memory
- Move Memory — allows you to move blocks of data in memory
- Examine Register — allows you to examine and modify the 8085's register contents
- GO — allows you to execute user programs

In addition to detailed information on using the monitors, the SDK-85 User's Manual provides circuit diagrams, a monitor listing, and a description of how the system works.



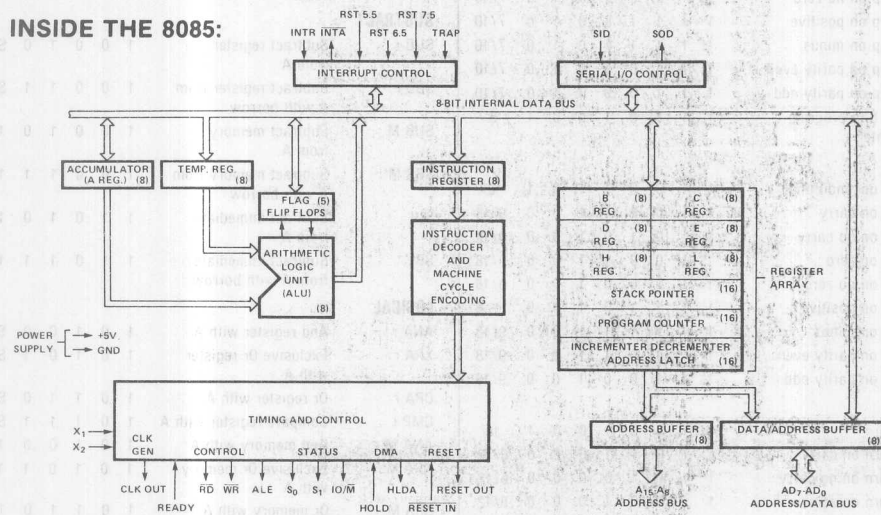
SDK-85 FUNCTIONAL BLOCK DIAGRAM

The SDK-85 is designed around Intel's 8085 Microprocessor. The Intel® 8085 is a new generation, complete 8-bit parallel central processing unit (CPU). Its instruction set is 100% software upward compatible with the 8080A microprocessor, and it is designed to improve the present 8080's performance by higher system speed. Its high level of system integration allows a minimum system of three IC's: 8085 (CPU), 8156 (RAM) and 8355/8755 (ROM/PROM).

The 8085 incorporates all of the features that the 8224 (clock generator) and 8228 (system controller) provided for the 8080, thereby offering a high level of system integration.

The 8085 uses a multiplexed Data Bus. The address is split between the 8-bit address bus and the 8-bit data bus. The on-chip address latches of 8155/8156/8355/8755 memory products allows a direct interface with 8085.

### INSIDE THE 8085:



- SEVEN 8-BIT REGISTERS. SIX OF THEM CAN BE LINKED IN REGISTER PAIRS FOR CERTAIN OPERATIONS.
- 8-BIT ALU.
- 16-BIT STACK POINTER (STACK IS MAINTAINED OFFBOARD IN SYSTEM RAM MEMORY).
- 16-BIT PROGRAM COUNTER.

### 8085 INSTRUCTION SET Summary of Processor Instructions

Mnemonic	Description	Instruction Code[1]								Clock[2] Cycles
		D <sub>7</sub>	D <sub>6</sub>	D <sub>5</sub>	D <sub>4</sub>	D <sub>3</sub>	D <sub>2</sub>	D <sub>1</sub>	D <sub>0</sub>	
MOVE, LOAD, AND STORE										
MOV r <sub>1</sub> , r <sub>2</sub>	Move register to register	0	1	D	D	D	S	S	S	4
MOV M, r	Move register to memory	0	1	1	1	0	S	S	S	7
MOV r, M	Move memory to register	0	1	D	D	D	1	1	0	7
MVI r	Move immediate register	0	0	D	D	D	1	1	0	7
MVI M	Move immediate memory	0	0	1	1	0	1	1	0	10
LXI B	Load immediate register Pair B & C	0	0	0	0	0	0	0	1	10
LXI D	Load immediate register Pair D & E	0	0	0	1	0	0	0	1	10
LXI H	Load immediate register Pair H & L	0	0	1	0	0	0	0	1	10
STAX B	Store A indirect	0	0	0	0	0	0	1	0	7
STAX D	Store A indirect	0	0	0	1	0	0	1	0	7
LDAX B	Load A indirect	0	0	0	0	1	0	1	0	7
LDAX D	Load A indirect	0	0	0	1	1	0	1	0	7
STA	Store A direct	0	0	1	1	0	0	1	0	13
LDA	Load A direct	0	0	1	1	1	0	1	0	13
SHLD	Store H & L direct	0	0	1	0	0	0	1	0	16
LHLD	Load H & L direct	0	0	1	0	1	0	1	0	16
XCHG	Exchange D & E, H & L Registers	1	1	1	0	1	0	1	1	4

Mnemonic	Description	Instruction Code[1]								Clock[2] Cycles
		D <sub>7</sub>	D <sub>6</sub>	D <sub>5</sub>	D <sub>4</sub>	D <sub>3</sub>	D <sub>2</sub>	D <sub>1</sub>	D <sub>0</sub>	
STACK OPS										
PUSH B	Push register Pair B & C on stack	1	1	0	0	0	1	0	1	12
PUSH D	Push register Pair D & E on stack	1	1	0	1	0	1	0	1	12
PUSH H	Push register Pair H & L on stack	1	1	1	0	0	1	0	1	12
PUSH PSW	Push A and Flags on stack	1	1	1	1	0	1	0	1	12
POP B	Pop register Pair B & C off stack	1	1	0	0	0	0	0	1	10
POP D	Pop register Pair D & E off stack	1	1	0	1	0	0	0	1	10
POP H	Pop register Pair H & L off stack	1	1	1	0	0	0	0	1	10
POP PSW	Pop A and Flags off stack	1	1	1	1	0	0	0	1	10
XTHL	Exchange top of stack: H & L	1	1	1	0	0	0	1	1	16
SPHL	H & L to stack pointer	1	1	1	1	1	0	0	1	6
LXI SP	Load immediate stack pointer	0	0	1	1	0	0	0	1	10
INX SP	Increment stack pointer	0	0	1	1	0	0	1	1	6
DCX SP	Decrement stack pointer	0	0	1	1	1	0	1	1	6

# 8085 INSTRUCTION SET Summary of Processor Instructions (Cont.)

Mnemonic	Description	Instruction Code[1]								Clock[2]
		D <sub>7</sub>	D <sub>6</sub>	D <sub>5</sub>	D <sub>4</sub>	D <sub>3</sub>	D <sub>2</sub>	D <sub>1</sub>	D <sub>0</sub>	
<b>JUMP</b>										
JMP	Jump unconditional	1	1	0	0	0	0	1	1	10
JC	Jump on carry	1	1	0	1	1	0	1	0	7/10
JNC	Jump on no carry	1	1	0	1	0	0	1	0	7/10
JZ	Jump on zero	1	1	0	0	1	0	1	0	7/10
JNZ	Jump on no zero	1	1	0	0	0	0	1	0	7/10
JP	Jump on positive	1	1	1	1	0	0	1	0	7/10
JM	Jump on minus	1	1	1	1	1	0	1	0	7/10
JPE	Jump on parity even	1	1	1	0	1	0	1	0	7/10
JPO	Jump on parity odd	1	1	1	0	0	0	1	0	7/10
PCHL	H & L to program counter	1	1	1	0	1	0	0	1	6
<b>CALL</b>										
CALL	Call unconditional	1	1	0	0	1	1	0	1	18
CC	Call on carry	1	1	0	1	1	1	0	0	9/18
CNC	Call on no carry	1	1	0	1	0	1	0	0	9/18
CZ	Call on zero	1	1	0	0	1	1	0	0	9/18
CNZ	Call on no zero	1	1	0	0	0	1	0	0	9/18
CP	Call on positive	1	1	1	1	0	1	0	0	9/18
CM	Call on minus	1	1	1	1	1	1	0	0	9/18
CPE	Call on parity even	1	1	1	0	1	1	0	0	9/18
CPO	Call on parity odd	1	1	1	0	0	1	0	0	9/18
<b>RETURN</b>										
RET	Return	1	1	0	0	1	0	0	1	10
RC	Return on carry	1	1	0	1	1	0	0	0	6/12
RNC	Return on no carry	1	1	0	1	0	0	0	0	6/12
RZ	Return on zero	1	1	0	0	1	0	0	0	6/12
RNZ	Return on no zero	1	1	0	0	0	0	0	0	6/12
RP	Return on positive	1	1	1	1	0	0	0	0	6/12
RM	Return on minus	1	1	1	1	1	0	0	0	6/12
RPE	Return on parity even	1	1	1	0	1	0	0	0	6/12
RPO	Return on parity odd	1	1	1	0	0	0	0	0	6/12
<b>RESTART</b>										
RST	Restart	1	1	A	A	A	1	1	1	12
<b>INCREMENT AND DECREMENT</b>										
INR r	Increment register	0	0	D	D	D	1	0	0	4
DCR r	Decrement register	0	0	D	D	D	1	0	1	4
INR M	Increment memory	0	0	1	1	0	1	0	0	10
DCR M	Decrement memory	0	0	1	1	0	1	0	1	10
INX B	Increment B & C registers	0	0	0	0	0	0	1	1	6
INX D	Increment D & E registers	0	0	0	1	0	0	1	1	6
INX H	Increment H & L registers	0	0	1	0	0	0	1	1	6
DCX B	Decrement B & C	0	0	0	0	1	0	1	1	6
DCX D	Decrement D & E	0	0	0	1	1	0	1	1	6
DCX H	Decrement H & L	0	0	1	0	1	0	1	1	6
<b>ADD</b>										
ADD r	Add register to A	1	0	0	0	0	S	S	S	4
ADC r	Add register to A with carry	1	0	0	0	1	S	S	S	4
ADD M	Add memory to A	1	0	0	0	0	1	1	0	7
ADC M	Add memory to A with carry	1	0	0	0	1	1	1	0	7
ADI	Add immediate to A	1	1	0	0	0	1	1	0	7
ACI	Add immediate to A with carry	1	1	0	0	1	1	1	0	7
<b>DAD</b>										
DAD B	Add B & C to H & L	0	0	0	0	1	0	0	1	10
DAD D	Add D & E to H & L	0	0	0	1	1	0	0	1	10
DAD H	Add H & L to H & L	0	0	1	0	1	0	0	1	10
DAD SP	Add stack pointer to H & L	0	0	1	1	1	0	0	1	10
<b>SUBTRACT</b>										
SUB r	Subtract register from A	1	0	0	1	0	S	S	S	4
SBB r	Subtract register from A with borrow	1	0	0	1	1	S	S	S	4
SUB M	Subtract memory from A	1	0	0	1	0	1	1	0	7
SBB M	Subtract memory from A with borrow	1	0	0	1	1	1	1	0	7
SUI	Subtract immediate from A	1	1	0	1	0	1	1	0	7
SBI	Subtract immediate from A with borrow	1	1	0	1	1	1	1	0	7
<b>LOGICAL</b>										
ANA r	And register with A	1	0	1	0	0	S	S	S	4
XRA r	Exclusive Or register with A	1	0	1	0	1	S	S	S	4
ORA r	Or register with A	1	0	1	1	0	S	S	S	4
CMP r	Compare register with A	1	0	1	1	1	S	S	S	4
ANA M	And memory with A	1	0	1	0	0	1	1	0	7
XRA M	Exclusive Or memory with A	1	0	1	0	1	1	1	0	7
ORA M	Or memory with A	1	0	1	1	0	1	1	0	7
CMP M	Compare memory with A	1	0	1	1	1	1	1	0	7
ANI	And immediate with A	1	1	1	0	0	1	1	0	7
XRI	Exclusive Or immediate with A	1	1	1	0	1	1	1	0	7
ORI	Or immediate with A	1	1	1	1	0	1	1	0	7
CPI	Compare immediate with A	1	1	1	1	1	1	1	0	7
<b>ROTATE</b>										
RLC	Rotate A left	0	0	0	0	0	1	1	1	4
RRC	Rotate A right	0	0	0	0	1	1	1	1	4
RAL	Rotate A left through carry	0	0	0	1	0	1	1	1	4
RAR	Rotate A right through carry	0	0	0	1	1	1	1	1	4
<b>SPECIALS</b>										
CMA	Complement A	0	0	1	0	1	1	1	1	4
STC	Set carry	0	0	1	1	0	1	1	1	4
CMC	Complement carry	0	0	1	1	1	1	1	1	4
DAA	Decimal adjust A	0	0	1	0	0	1	1	1	4
<b>INPUT/OUTPUT</b>										
IN	Input	1	1	0	1	1	0	1	1	10
OUT	Output	1	1	0	1	0	0	1	1	10
<b>CONTROL</b>										
EI	Enable Interrupts	1	1	1	1	1	0	1	1	4
DI	Disable Interrupt	1	1	1	1	0	0	1	1	4
NOP	No-operation	0	0	0	0	0	0	0	0	4
HLT	Halt	0	1	1	1	0	1	1	0	5
<b>NEW 8085 INSTRUCTIONS</b>										
RIM	Read Interrupt Mask	0	0	1	0	0	0	0	0	4
SIM	Set Interrupt Mask	0	0	1	1	0	0	0	0	4

NOTES: 1. DDD or SSS. B=000, C=001, D 010, E 011, H 100, L 101, Memory=110, A-111.

2. Two possible cycle times, (6/12) indicate instruction cycles dependent on condition flags.

\*All mnemonics copyright

©Intel Corporation 1977

## SDK-85 SPECIFICATIONS

### Central Processor

CPU: 8085  
Instruction Cycle: 1.3 microsecond  
Tcy: 330 ns

### Memory

ROM: 2K bytes (expandable to 4K bytes) 8355/8755  
RAM: 256 bytes (expandable to 512 bytes) 8155  
Addressing:  
ROM 0000-07FF (expandable to 0FFF with an additional 8355/8755)  
RAM 2000-20FF (2800-28FF available with an additional 8155)

Note: The wire-wrap area of the SDK-85 PC board may be used for additional custom memory expansion up to the 64K byte addressing limit of the 8085.

### Input/Output

Parallel: 38 lines (expandable to 76 lines).  
Serial: Through SID/SOD ports of 8085. Software generated baud rate.  
Baud Rate: 110

### Interfaces

Bus: All signals TTL compatible.  
Parallel I/O: All signals TTL compatible.  
Serial I/O: 20 mA current loop TTY

Note: By populating the buffer area of the board, the user has access to all bus signals which enable him to design custom system expansions into the kit's wire-wrap area.

### Interrupts

Three Levels: (RST 7.5) — Keyboard Interrupt.  
(RST 6.5) — TTL Input  
(INTR) — TTL Input

### DMA

Hold Request: Jumper selectable. TTL compatible input.

### Software

System Monitor: Pre-programmed 8755 or 8355 ROM  
Addresses: 0000-07FF.  
Monitor I/O: Keyboard/Display or TTY (serial I/O)

### Literature

Design Library (Provided with kit):

- SDK-85 User's Manual
- MCS-85 User's Manual
- 8080/8085 Assembly Language Programming Manual
- Intellec® MDS Brochure
- ICE-85 Data Sheet
- PL/M-80 Data Sheet
- 8085/8080 Assembly Language Reference Card

### Physical Characteristics

Width: 12.0 in.  
Height: 10 in.  
Depth: 0.50 in.  
Weight: approx. 12 oz.

### Electrical Characteristics (DC Power Required — Power Supply Not Included in Kit)

V <sub>CC</sub> 5V ±5%	1.3 Amps	
V <sub>TTY</sub> -10V ± 10%	0.3 Amps	(V <sub>TTY</sub> required only if teletype is connected)

### Environmental

Operating Temperature: 0-55°C







## **FORTRAN-80 8080/8085 ANS FORTRAN 77 INTELLEC® RESIDENT COMPILER**

**Meets and exceeds ANS FORTRAN 77 Subset Language Specification**

**Supports Intel Floating Point Standard**

**Resident operation on Intellec® Microcomputer Development System and Intellec® Series II Microcomputer Development System**

**Supports full symbolic debugging with ICE-80 and ICE-85**

**Produces relocatable and linkable object code compatible with resident PL/M-80 and 8080/8085 Macro Assembler**

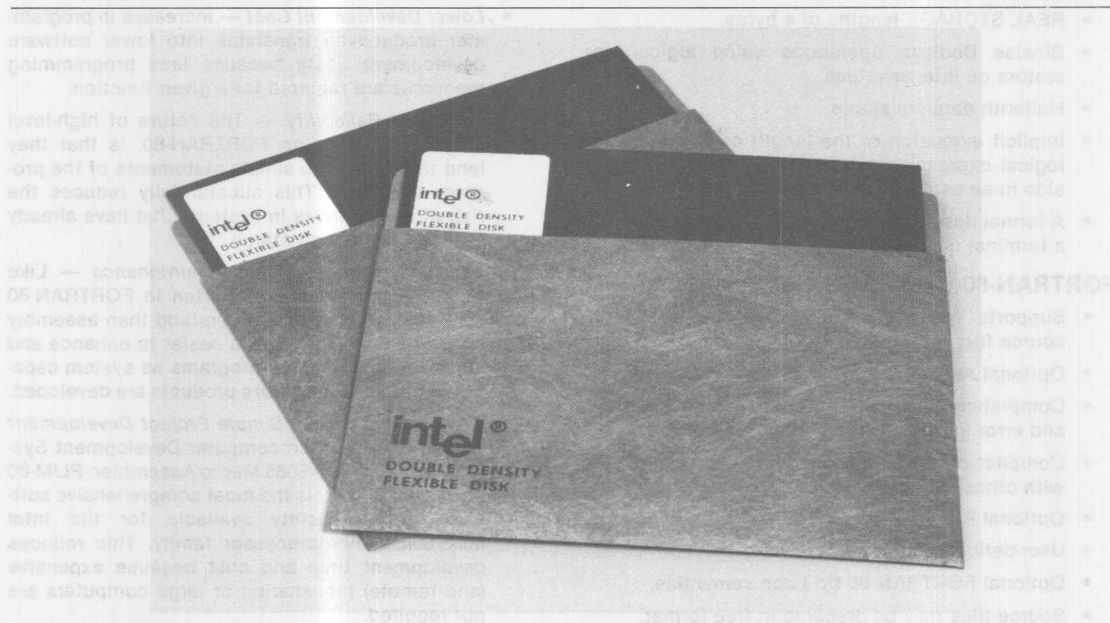
**Full FORTRAN 77 language I/O support when used with ISIS-II run-time library**

**Sophisticated code optimization insures efficient program implementation**

FORTRAN-80 is a computer industry-standard, high-level programming language and compiler that translates FORTRAN statements into relocatable object modules. When the object modules are linked together and located into absolute program modules, they are suitable for execution on Intel® 8080/8085 Microprocessors, ISBC-80 OEM Computer Systems, and Intellec® Microcomputer Development Systems. FORTRAN-80 meets and exceeds the ANS FORTRAN 77 Language Subset Specification<sup>1</sup>. The compiler operates on the Intellec Microcomputer Development System under the ISIS-II Disk Operating Systems and produces efficient relocatable object modules that are compatible for linkage with PL/M-80 and 8080/8085 Macro Assembler modules.

The ANS FORTRAN 77 language specification offers many powerful extensions to the FORTRAN language that are especially well suited to Intel® 8080/8085 Microprocessor software development. Because FORTRAN-80 conforms to the ANS FORTRAN 77 standard, the user is assured of compatibility with existing FORTRAN software that meets the standard as well as a guarantee of upward compatibility to other computer systems supporting an ANS FORTRAN 77 Compiler.

<sup>1</sup>ANSI X3J3/90



## FORTRAN-80 LANGUAGE FEATURES

Major ANS FORTRAN 77 features supported by the Intel® FORTRAN-80 Programming Language include:

- Structured Programming is supported with the IF... THEN... ELSE IF... ELSE... END IF constructs.
- CHARACTER data type permits alphanumeric data to be handled as strings rather than characters stored in array elements.
- Full I/O capabilities include:
  - Sequential and Direct Access files
  - Error handling facilities
  - Formatted, Free-formatted, and Unformatted data representation
  - Internal (in-memory) file units provide capability to format and reformat data in internal memory buffers
  - List Directed Formatting
- Supports arrays of up to seven dimensions.
- Supports logical operators
  - .EQV. — Logical equivalence
  - .NEQV. — Logical nonequivalence

Major extensions to FORTRAN 77 in Intel FORTRAN-80 include:

- Direct 8080/8085 port I/O supported by intrinsic subroutines.
- Binary and Hexadecimal integer constants.
- User-defined INTEGER storage lengths of 1, 2 or 4 bytes.
- User-defined LOGICAL storage lengths of 1, 2 or 4 bytes.
- REAL STORAGE lengths of 4 bytes.
- Bitwise Boolean operations using logical operators on integer values.
- Hollerith data constants.
- Implicit extension of the length of an integer or logical expression to the length of the left-hand side in an assignment statement.
- A format descriptor to suppress carriage return on a terminal output device at the end of the record.

## FORTRAN-80 COMPILER FEATURES

- Supports multiple compilation units in single source file.
- Optional Assembly Language code listing.
- Comprehensive cross-reference, symbol attribute and error listing.
- Compiler controls and directives are compatible with other Intel language translators.
- Optional Reentrancy.
- User-defined default storage lengths.
- Optional FORTRAN 66 Do Loop semantics.
- Source files may be prepared in free format.

- The INCLUDE control permits specified source files to be combined into a compilation unit at compile time.

## FORTRAN-80 BENEFITS

FORTRAN-80 provides a means of developing application software for Intel® MCS-80/85 products in a familiar, widely accepted, and computer industry-standardized programming language. FORTRAN-80 will greatly enhance the user's ability to provide cost-effective solutions to software development for Intel microprocessors as illustrated by the following:

- *Completely Complementary to Existing Intel Software Design Tools* — Object modules are linkable with new or existing Assembly Language and PL/M Modules.
- *Incremental Runtime Library Support* — Runtime overhead is limited only to facilities required by the program.
- *Low Learning Effort* — FORTRAN-80, like PL/M, is easy to learn and use. Existing FORTRAN software can be ported to FORTRAN-80, and programs developed in FORTRAN-80 can be run on any other computer with ANS FORTRAN 77.
- *Earlier Project Completion* — Critical projects are completed earlier than otherwise possible because FORTRAN-80 will substantially increase programmer productivity, and is complementary to PL/M Modules by providing comprehensive arithmetic, I/O formatting, and data management support in the language.
- *Lower Development Cost* — Increases in programmer productivity translates into lower software development costs because less programming resources are required for a given function.
- *Increased Reliability* — The nature of high-level languages, including FORTRAN-80, is that they lend themselves to simple statements of the program algorithm. This substantially reduces the risk of costly errors in systems that have already reached production status.
- *Easier Enhancements and Maintenance* — Like PL/M, program modules written in FORTRAN-80 are easier to read and understand than assembly language. This means it is easier to enhance and maintain FORTRAN-80 programs as system capabilities expand and future products are developed.
- *Comprehensive, Yet Simple Project Development* — The Intel Microcomputer Development System, with the 8080/8085 Macro Assembler, PL/M-80 and FORTRAN-80 is the most comprehensive software design facility available for the Intel MCS-80/85 Microprocessor family. This reduces development time and cost because expensive (and remote) timesharing or large computers are not required.

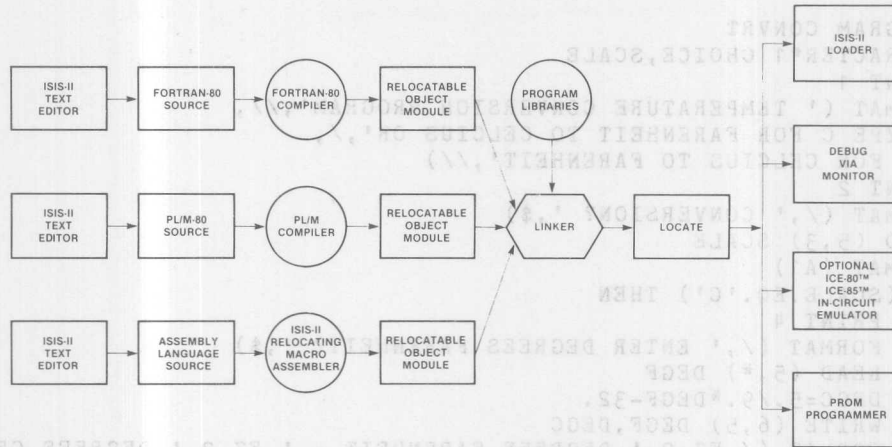
**SAMPLE FORTRAN-80 SOURCE PROGRAM LISTING**

```

C THIS PROGRAM IS AN EXAMPLE OF ISIS-II FORTRAN-80 THAT
C CONVERTS TEMPERATURE BETWEEN CELCIUS AND FARENHEIT
C
      PROGRAM CONVRT
      CHARACTER*1 CHOICE,SCALE
      PRINT 1
1     FORMAT (' TEMPERATURE CONVERSION PROGRAM',//,
      *' TYPE C FOR FARENHEIT TO CELCIUS OR',/,
      *' F FOR CELCIUS TO FARENHEIT',//)
10    PRINT 2
2     FORMAT (/, ' CONVERSION? ', $)
      READ (5,3) SCALE
3     FORMAT (A1)
      IF (SCALE.EQ.'C') THEN
4         PRINT 4
          FORMAT (/, ' ENTER DEGREES FARENHEIT? ', $)
          READ (5,*) DEGF
          DEGC=5./9.*DEGF-32.
          WRITE (6,5) DEGF,DEGC
5          FORMAT (/,F7.2,' DEGREES FARENHEIT = ',F7.2,' DEGREES CELCIUS',/)
11         PRINT 6
6         FORMAT (/, ' AGAIN (Y OR N)? ', $)
          READ (5,3) CHOICE
          IF (CHOICE.EQ.'Y') THEN
              GOTO 10
          ELSE IF (CHOICE.EQ.'N') THEN
              CALL EXIT
          ELSE
              GOTO 11
          END IF
      ELSE IF (SCALE.EQ.'F') THEN
7         PRINT 7
          FORMAT (/, ' ENTER DEGREES CELCIUS? ', $)
          READ (5,*) DEGC
          DEGF=9./5.*DEGC+32.
          WRITE(6,8) DEGC,DEGF
8         FORMAT (/,F7.2,' DEGREES CELCIUS = ',F7.2,' DEGREES FARENHEIT',/)
          GOTO 11
      ELSE
9         WRITE (6,9) SCALE
          FORMAT (/,1H ,A1,' NOT A VALID CHOICE - RETRY!',/)
          GOTO 10
      END IF
      END
  
```

## FORTRAN-80

The FORTRAN-80 Compiler is an efficient, multiphase compiler that accepts source programs, translates them into relocatable object code, and produces requested listings. After compilation, the object program may be linked to other modules, located to a specific area of memory, then executed. The diagram shown below illustrates a program development cycle where the program consists of modules created by FORTRAN-80, PL/M-80 and the 8080/8085 Macro Assembler.



### SPECIFICATIONS

#### OPERATING ENVIRONMENT

##### Required Hardware:

- Intellec® Microcomputer Development System
  - MDS-800, MDS-888
  - Series II Model 220, Model 230
- 64K bytes of RAM memory
- Dual diskette drives
  - Single or Double Density
- System console
  - CRT or hardcopy interactive device

##### Optional Hardware:

- Line Printer
- ICE-80™, ICE-85™

##### Required Software:

- ISIS-II Diskette Operating System
  - Single or Double Density

### DOCUMENTATION PACKAGE

- FORTRAN-80 Programming Manual (9800481)
- ISIS-II FORTRAN-80 Compiler Operator's Manual (9800480)
- FORTRAN-80 Programming Reference Card (9800547)

### SHIPPING MEDIA

- Flexible Diskettes
  - Single and Double Density

# APPENDIX 1

## Applications of MCS-85

**MCS  
85**  
**MCS  
85**  
**MCS  
85**

# Appendix 1

## Applications of MCS-85

### Contents

#### SECTION 1 INTRODUCTION TO MCS-85 APPLICATIONS

Introduction .....	A1-1
MCS-85™ System .....	A1-1
Sample Applications .....	A1-1
Baud Rate Generator .....	A1-1
Serial Communications .....	A1-1
Small System .....	A1-6
Block Move, Block Search .....	A1-6
RST 7 .....	A1-7

#### SECTION 2 DETAILED APPLICATION EXAMPLES

Memory Addressing .....	A1-8
ROM, EPROM .....	A1-8
Static RAM .....	A1-11
Dynamic RAM .....	A1-11
<b>System Timings</b> .....	A1-16
Clock In vs. Clock Out vs. Control Timings .....	A1-16
3.125 vs. 5 MHz Timings .....	A1-18
Memory Device Compatibility .....	A1-20
Minimum System .....	A1-20
Medium System .....	A1-20
Follow Through of Dynamic RAM Example .....	A1-22
Peripheral Device Compatibility .....	A1-23
Loading, Decoupling Considerations .....	A1-25

#### APPLICATION EXAMPLE 1

<b>Minimum System Application Example as a Temperature Sensor</b> .....	A1-26
Application .....	A1-26
8155 Counter Read Procedure .....	A1-28
Software Block Move, Search Illustrations .....	A1-29

#### APPLICATION EXAMPLE 2

<b>CRT Interface</b> .....	A1-32
Hardware Interface .....	A1-32
Software Package .....	A1-32
Output Routine .....	A1-34
Input Routine .....	A1-35
Timing Analysis .....	A1-35
Baud Rate Identification Routine .....	A1-36

#### APPLICATION EXAMPLE 3

<b>Cassette Recorder Interface</b> .....	A1-38
Hardware Design .....	A1-38
Software .....	A1-38
Output Routine .....	A1-40
Input Routine .....	A1-40
<b>Additional Comments</b> .....	A1-42
<b>Appendix 1.1 Temperature Sensor Code</b> .....	A1-44
<b>Appendix 1.2 CRT and Cassette Code</b> .....	A1-48

# APPENDIX 1 APPLICATIONS OF MCS-85™

## SECTION 1 INTRODUCTION TO MCS-85 APPLICATIONS

When the first microprocessor was introduced about five years ago, it was largely ignored by the electronics industry. However, since that inauspicious beginning, this new device has become the hottest topic in current technology. As more and more product designers become familiar with the capabilities of microcomputers, the number of new applications increases geometrically. In most of these applications, the new technology has been used to replace designs which were formerly implemented with TTL logic and under-utilized minicomputers. However, an increasing number of products are surfacing which would have been impractical prior to the microcomputer era.

Microcomputers are being applied to a wide range of data communications tasks. The field of telephone equipment is being invaded by systems which control and monitor calls. Point of sale terminals are increasing daily with the addition of interface to coin changers, electronic scales and remote computers. Small stand-alone computers are relying heavily upon microcomputers in teleprocessing, time-sharing, data base management and similar interactive applications. An increasing number of microcomputer based data terminals are providing local interactive intelligence with programmable character sets, vector generation and the pre-processing of data.

Instrumentation is widely utilizing the microprocessor for a variety of control and arithmetic processing functions. Microcomputers are controlling laboratory equipment such as oscilloscopes, DVM's, network analyzers and frequency synthesizers. Medical electronics are crediting microcomputers with tasks such as patient monitoring, blood analysis and X-ray scanning. Travel is becoming microcomputerized by automotive control, air and ocean navigation equipment and rapid transit systems.

### MCS-85™ SYSTEM

Many possible microcomputer applications have been overlooked because of the design tasks required to build the microcomputer. These tasks include the system clock, read/write memory, I/O ports, serial communications interface and bus control logic. The MCS-85 system will enable the design engineer to concentrate on the application of the microcomputer, rather than on the implementation details.

The MCS-85 is yet another family of components which has the potential to provide a solution to the three problems which will always plague designers: cost, size and power. The reduced component count of an MCS-85 microcomputer, coupled with the increased integration of functions reduces both cost and size while increasing power.

COUNT (DECIMAL)	B.UD. RAT.
10240	200
5120	400
2560	800
1280	1600
640	3200
320	6400

## MCS-85™ APPLICATIONS

Programmable Video Game  
Process Control System  
Line Printer

Digital Multimeter  
Graphic Terminal  
Automotive Control

Spectrum Analyzer  
Front End Processor  
Credit Verifier

Network Analyzer  
Frequency Synthesizer

APPLICATION	PERIPHERAL DEVICES ENCOUNTERED	MCS-85™ COMPONENTS	
Intelligent Terminals	Cathode Ray Tube Display	8275	8085A
	Printing Units	8155	8355
	Synchronous and Asynchronous data lines	8251	
	Cassette Tape Unit		
Gaming Machines	Keyboards	8279	
	Keyboards, pushbuttons and switches	8279	8085A
	Various display devices		8355
	Coin acceptors	8155	
Cash Registers	Coin dispensers		
	Keyboard or Input Switch Array	8279	8085A
	Change Dispenser	8155	8355
	Digital Display		
Accounting and Billing Machines	Ticket Printer		
	Magnetic Card reader		
	Communication interface	8273	
	Keyboard	8279	8085A
Telephone Switching Control	Printer Unit	8155	8355
	Cassette or other magnetic tape unit	8257	
	"Floppy" disks	8271	
	Telephone Line Scanner	8253	8085A
Numerically Controlled Machines	Analog Switching Network		8355
	Dial Registers	8155	
	Class of Service Parcel		
	Magnetic or Paper Tape Reader	8155	8085A
Process Control	Stepper Motors		8355
	Optical Shaft Encoders		
	Analog-to-Digital Converters	8155	8085A
	Digital-to-Analog Converters		8355
	Control Switches	8279	
	Displays		

### Baud Rate Generator

Shown in Figure 2 is a minimum system configuration with the 8156 timer output connected to an 8085 interrupt input.

This configuration allows convenient use of the timer as a baud rate generator. A 6.144 MHz crystal is used as the frequency control element of the 8085A, providing integral divisors for the standard baud rates (300, 600, 1200, 2400, 4800, 9600 baud). The timer is programmed with the appropriate divisor (Figure 1) for the selected baud rate resulting in one pulse on the timer output for each bit cell time. The clock output (CLK) of the 8085A is used to clock the timer (TIMERIN). The frequency of this clock is one-half the crystal frequency or in this example 3.072 MHz. **TIMEROUT** now provides a crystal controlled pulse train at the baud rate selected.

### Serial Communications

By feeding the **TIMEROUT** signal of the 8156 back to the edge triggered **RST7.5** input of the 8085A, the processor can be interrupt driven at

the required baud rate. As shown in Figure 1, the minimum system supports serial communications with only the addition of the send and receive interface circuits.

The **SID** (SERIAL INPUT DATA) line and the **SOD** (SERIAL OUTPUT DATA) line are connected directly to a TTY or RS232 interface circuit. Assuming inverted data at the **SID** input, a direct connection is made to the **RST6.5** input for detection of the start bit.

Additional insight into using the 8085's serial I/O lines in communications application can be found in Section 2 of this Appendix.

BAUD RATE	COUNT (DECIMAL)
300	10,240
600	5,120
1200	2,560
2400	1,280
4800	640
9600	320

**FIGURE 1**

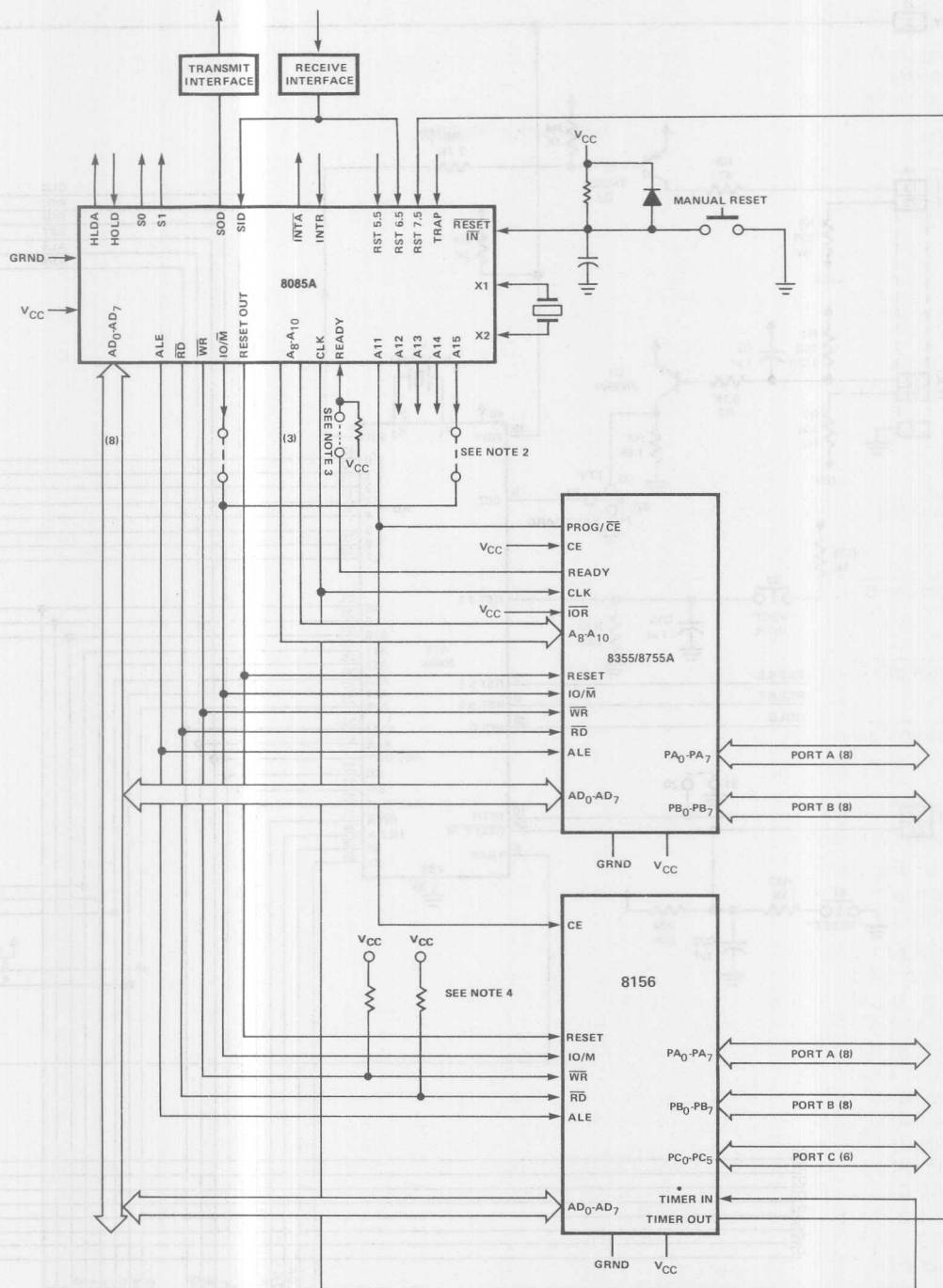


FIGURE 2 MINIMUM SYSTEM CONFIGURATION

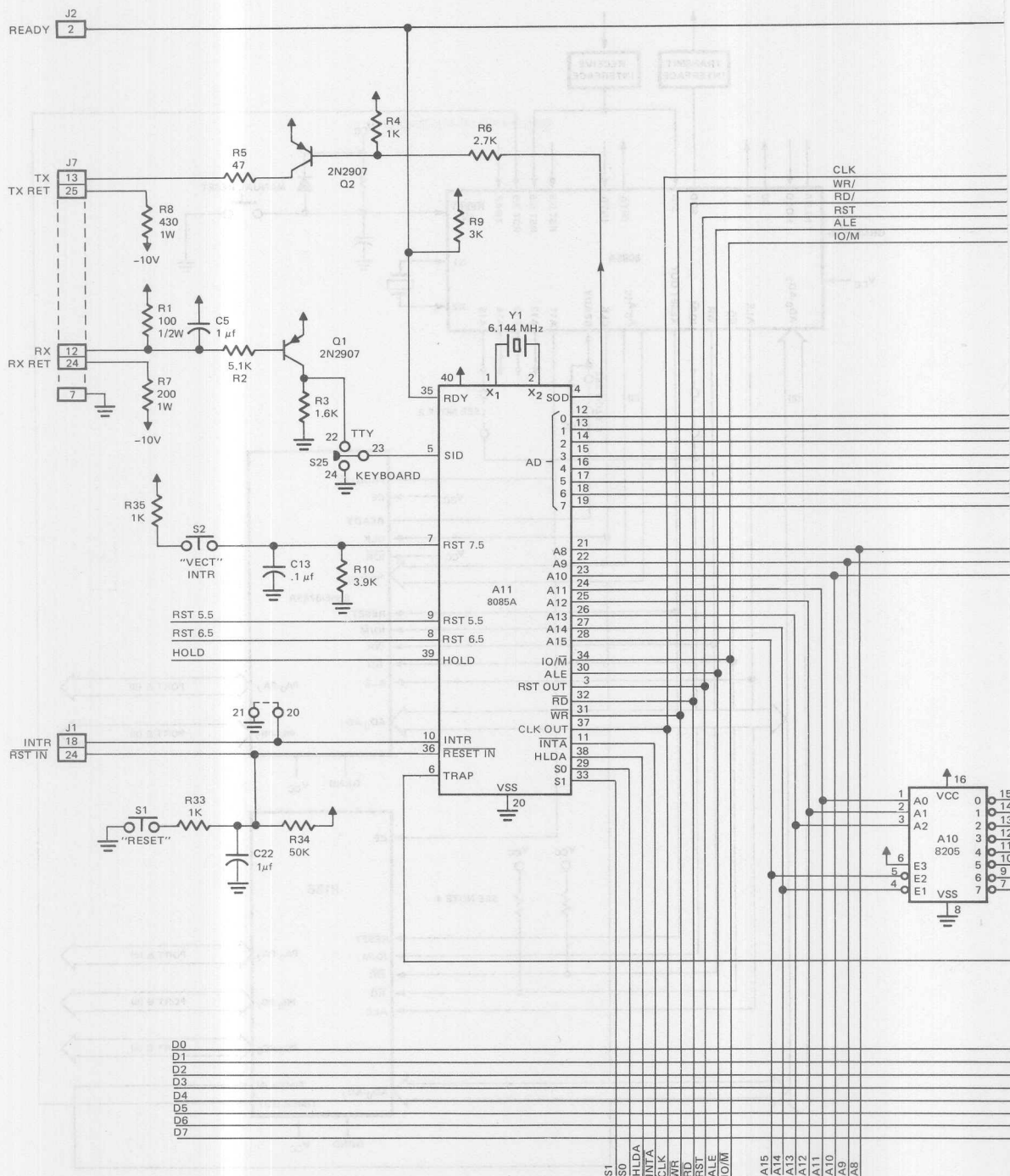
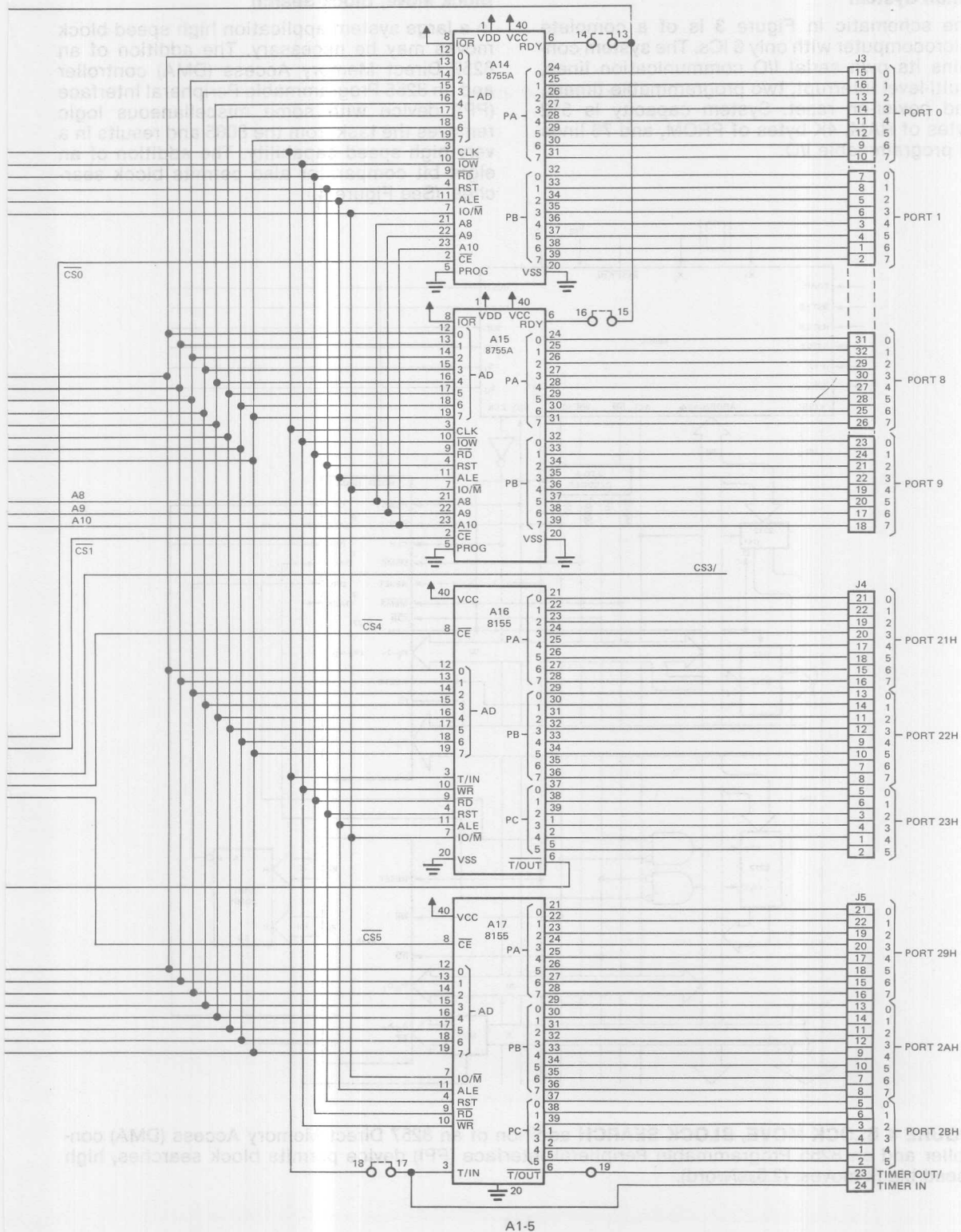


FIGURE 3 SMALL SYSTEM SCHEMATIC (similar to the schematic of Intel's SDK-85)

# MCS-85™ APPLICATIONS

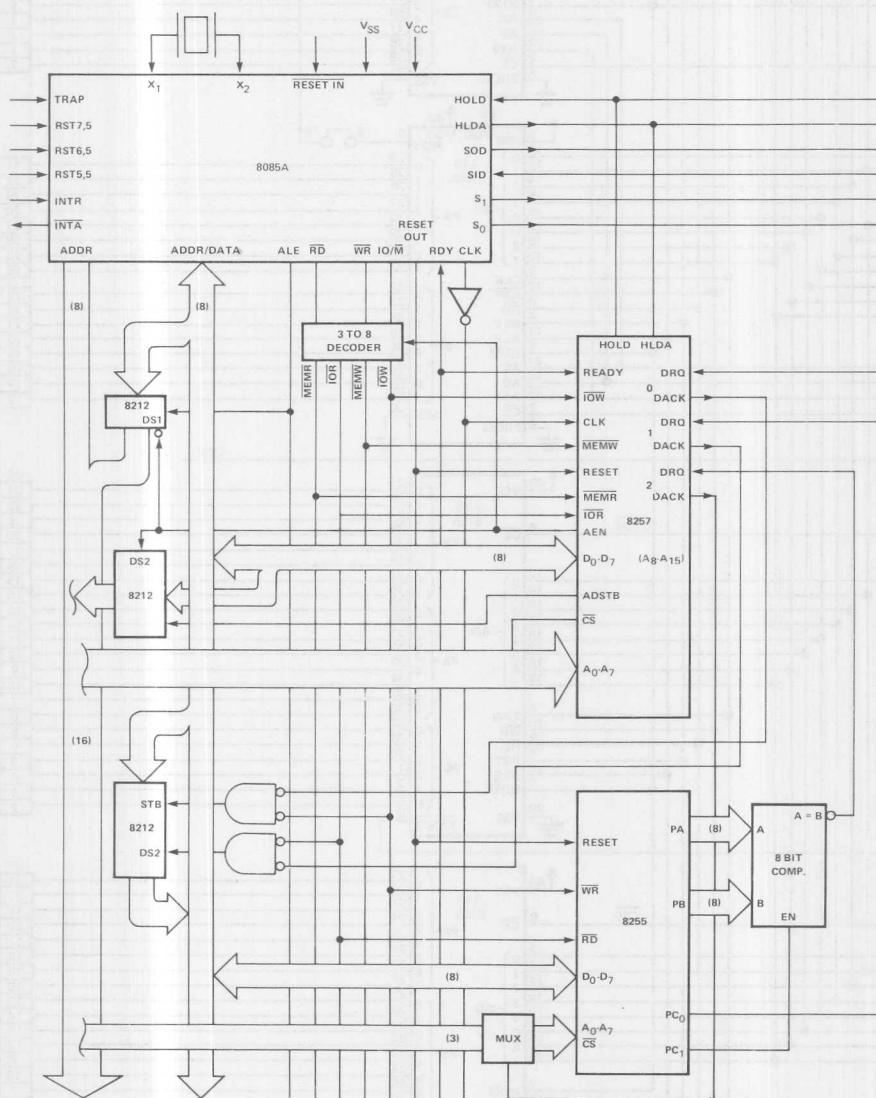


### Small System

The schematic in Figure 3 is of a complete microcomputer with only 6 ICs. The system contains its own serial I/O communication lines, multi-level interrupt, two programmable timers, and power-on reset. System capacity is 512 bytes of RAM, 4K bytes of PROM, and 76 lines of programmable I/O.

### Block Move, Block Search

In a large system application high speed block moves may be necessary. The addition of an 8257 Direct Memory Access (DMA) controller and an 8255 Programmable Peripheral Interface (PPI) device with some miscellaneous logic removes the task from the 8085 and results in a very high speed capability. The addition of an eight bit comparator also permits block searches. (See Figure 4.)



**FIGURE 4 BLOCK MOVE, BLOCK SEARCH** addition of an 8257 Direct Memory Access (DMA) controller and an 8255 Programmable Peripheral Interface (PPI) device permits block searches, high speed block moves. (2.5μs/word).

Basic operation, for a block move, is that the CPU loads the 8257 with the starting address of the source block and the length\* of the block into Channel 0. Channel 1 is programmed with the starting location of the destination block and the length. A bit in Port C of the 8255 is set by the CPU which initiates a DMA request on Channels 0 and 1. Because the 8257 is initialized to the rotating priority mode, the first DMA cycle is from Channel 0 which latches the data from the first location of the source block into the 8212. The second cycle will be from Channel 1 which will store the latched data into the first location of the destination block. The next cycle will return to Channel 0 and the sequence will start over again until the length (terminal count) is reached. Programming the 8257 stop bit insures that each channel will be disabled when its respective terminal count is reached.

This configuration also supports a block fill. DMA Channel 0 points to a location containing the fill value and has a length of one. Channel 1 points to the starting location of the destination block and contains the length. When the sequence is initiated the value will be loaded into the latch by Channel 0. Channel 0 reaches TC and is disabled. Priority rotates to Channel 1 which will repeatedly write into the destination block the value stored in the latch until TC is reached.

Block search operations use the 8-bit comparator and Ports A & B of the 8255 and Channel 2 of the 8257. The CPU loads Port B with the search value and the DMA channel with the search area (starting address and length). A Port C bit initiates the DMA READ request. Channel 2 DMA Acknowledge sets Port A of the 8255 up as the receiver for the DMA READ cycle by multiplexing A<sub>0</sub>, A<sub>1</sub>, and CS. Each cycle of the DMA then loads Port A with the value of the

\* (The value loaded into the low-order 14-bits of the terminal count register specifies the number of DMA cycles minus one before the Terminal Count (TC) output is activated. For instance, a terminal count of 0 would cause the TC output to be active in the first DMA cycle for that channel. In general, if Length = the number of desired DMA cycles, load the value Length-1 into the low-order 14-bits of the terminal count register.)

pointed-to location in the block. When Port A equals Port B, the output of the comparator will gate off the DMA request. The requesting program can now read the Channel 2 address which is pointing to the search value plus one. However, if the status register of the 8257 indicates that TC of Channel 2 has been reached, then no match was found.

## RST 7

On the 8080A/8228 system if one tied  $\overline{\text{INTA}}$  out of the 8228 to +12 volts through a 1K $\Omega$  resistor, the 8228 would generate a RST 7 instruction to the 8080A upon interrupt. This was a very inexpensive mechanism.

The 8085A has expanded this facility with the RST 5.5, 6.5, 7.5 inputs but is not compatible with the RST 7 generated by the 8228. (Figure 5) To maintain this compatibility it can be achieved by adding an 8212 which will force a RST 7 instruction into the bus upon interrupt acknowledge ( $\overline{\text{INTA}}$ ). (Figure 6)

RESTART	VECTOR LOCATION
RST 7	38 <sub>16</sub>
RST 5.5	2C <sub>16</sub>
RST 6.5	34 <sub>16</sub>
RST 7.5	3C <sub>16</sub>
TRAP	24 <sub>16</sub>

FIGURE 5

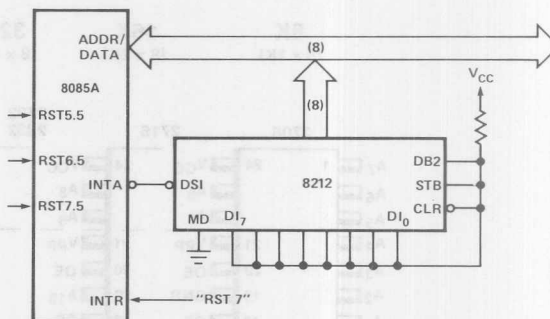


FIGURE 6

## SECTION 2 DETAILED APPLICATION EXAMPLES

### Memory Addressing

One of the necessary functions of the microprocessor bus is to interface with the memory where the program is stored. ROM and EPROM memories are typically used to store programs while static and dynamic RAMS are generally used for data memory. The following discussions cover the interfacing to be used for these types of memory.

### ROM - EPROM ADDRESSING

Later in this Appendix a section is devoted to an approach for developing a chart showing memory device compatibility for the 8085A. However, there is one area not included that will be discussed here, that is, unbuffered interfacing to standard ROM or EPROM memories. To use an unbuffered interface to ROM or EPROM it is necessary to understand a particular characteristic of the 8085A.

The 8085A has a period of time, T<sub>4</sub> through T<sub>6</sub> of the op code fetch cycle and certain instructions, where addresses A<sub>8</sub> through A<sub>15</sub> are undefined. Be careful about this. Not having addresses stable and using an address select method that would randomly turn on memory devices will cause bus contention and reliability problems in the unbuffered system. In the memory compatibility section of this Application Note, a minimum (unbuffered MCS-85 family and medium system (at least one level of buffering) configurations are considered. These configurations do not have bus contention problems. In the minimum system only MCS-85 components will be discussed where addresses are latched on the falling edge of ALE, thus ignoring any extraneous address transitions. The medium system is assumed to have data buffers that are enabled only at the proper time, thus again preventing any

bus contention problems. What about the user who wants to use standard ROM or EPROM without buffering?

As an example let's look at Intel's ROM/EPROM family (Fig. 3) and develop a system block diagram. This system should allow upward compatibility for these particular devices and avoid any bus contentions due to undefined addresses. In Figure 4 a traditional decoding scheme is shown that uses the time difference between t<sub>acc</sub> (address access) and t<sub>co</sub> (chip select access) to allow for decoding of the EPROM/ROM to be selected. Connecting only these signals, however, in an unbuffered system will result in data contention because of the spurious addresses during opcode fetch. The proper interconnect for this type of interface is shown in Figure 5 where an output enable ( $\overline{OE}$ ) signal will prevent any bus contention. This output enable is controlled by the read control signal,  $\overline{RD}$ , of the 8085A. This signal only occurs after addresses have stabilized.\*

Note also that a PROM is recommended for the decoding function vs. an 8205 (1 of 8 decoder). Why? This PROM allows the user to easily upgrade his system to the 32 and 64K versions with minimum rewiring. As seen in Figure 3, only 4 pins are being altered (18-21) in the Intel ROM/EPROM family to allow for this upward compatibility. All a user would need to do is initially design his layout for 28 pin devices, thereby allowing total flexibility from 8K through 64K with the ease of only changing a decoding PROM and a few wires.†

\* Both  $\overline{RD}$  and  $\overline{WR}$  signals should be pulled up to +5V through a resistor to avoid random selection during 3-state.

† Another method is shown later in Figure 15 that facilitates the use of a decoder, such as the Intel 8205.

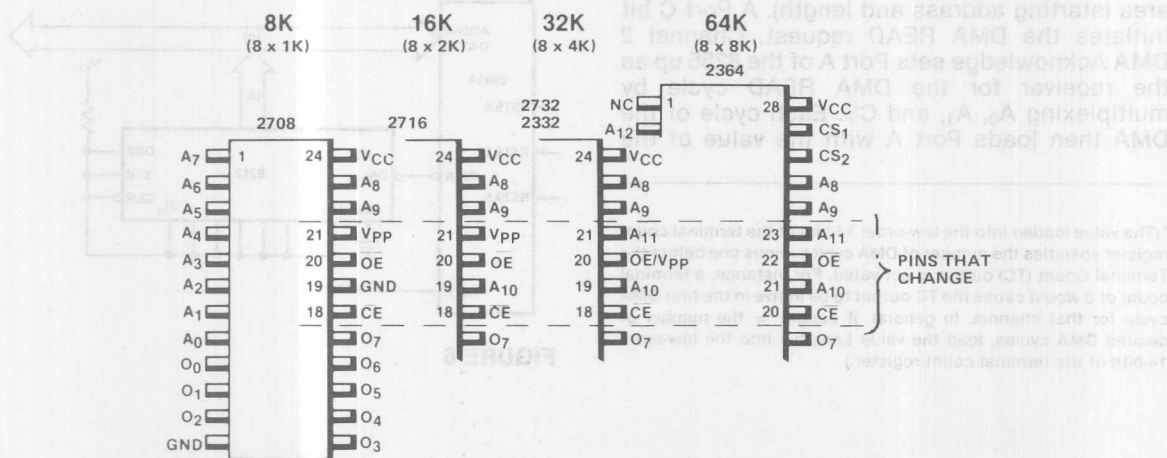


Figure 3. Intel EPROM/ROM Compatible Family.

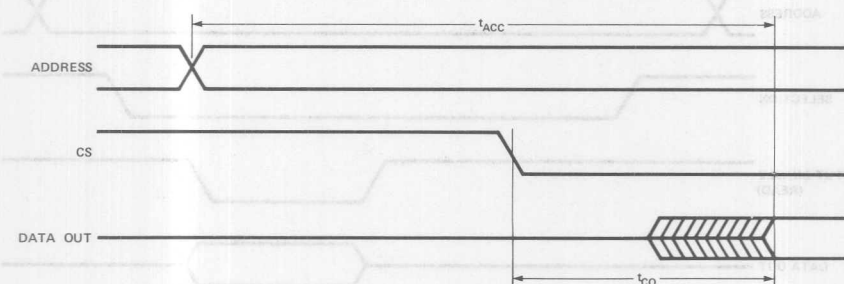
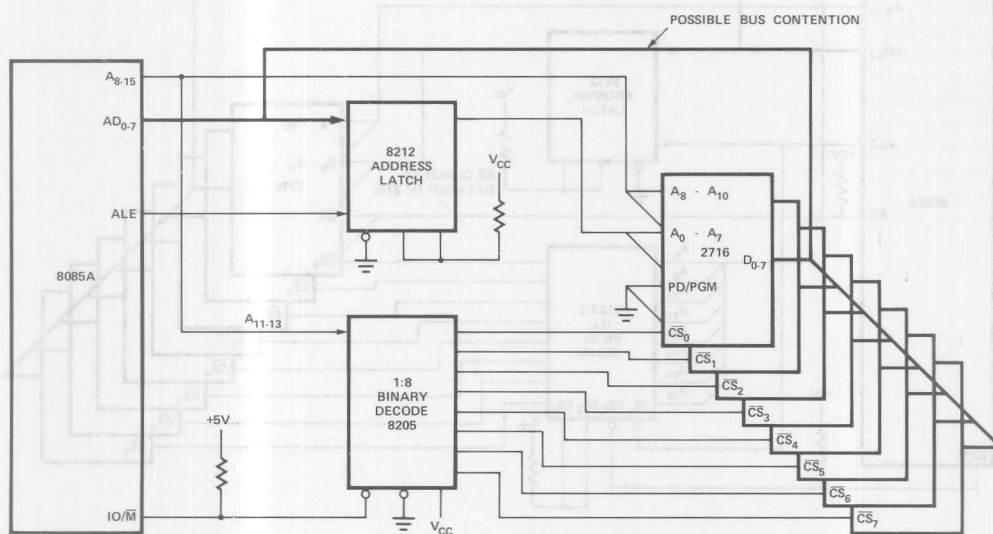
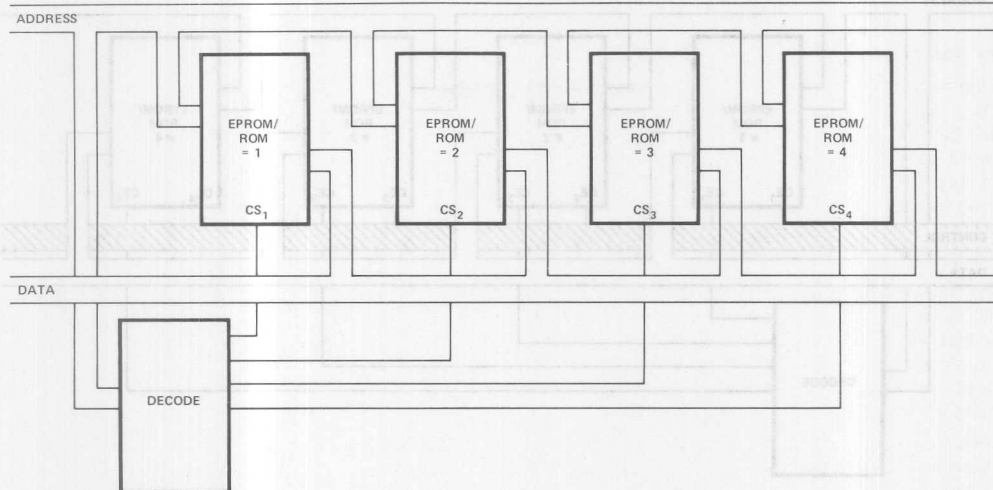


Figure 4. Traditional 16K EPROM System.



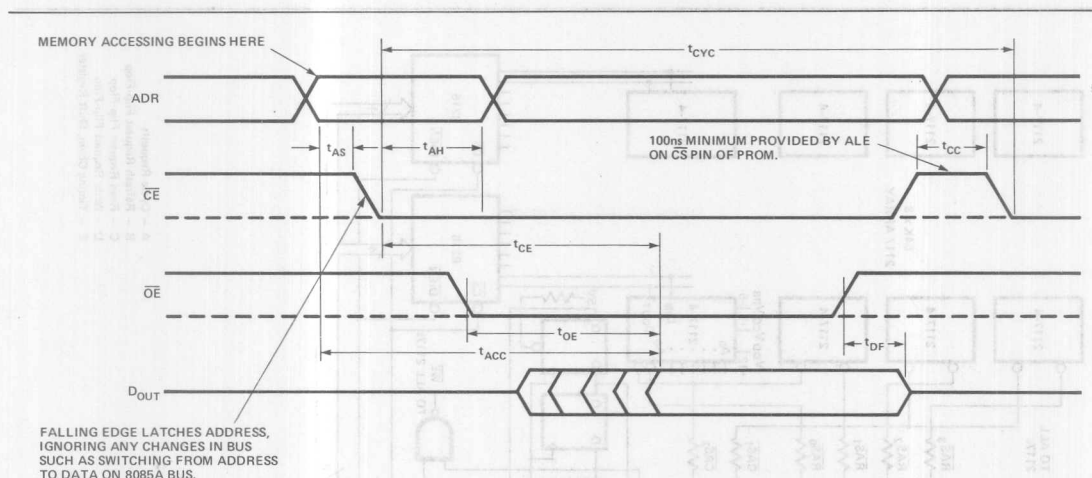


Figure 6. Edge Enabled Product Waveforms.

Another interesting aspect with this configuration is when working with the 2332, (4K x 8 ROM) or 2364 (8K x 8 ROM) the 8212 Demultiplexer no longer becomes necessary. In other words, you don't have to demultiplex the 8085A bus! To understand this it will be helpful to look at Figure 6. The new ROMs, 2332 and 2364, use an Edge Enabled concept allowing the addresses to be latched on the CE input. This approach allows access time to start as soon as addresses are valid. To latch another set of addresses CE must go high for at least 100 ns ( $t_{CC}$ ) and then go low again. This 100 ns can be accounted for by connecting ALE from the 8085A to the Cs1 input pin of the 3604A PROM device in the block diagrams shown. This assures that all devices are deselected for at least 100 ns while addresses are changing in the 8085A system.

#### STATIC MEMORIES

The same consideration must be applied to standard static memories as with the ROMs/EPROMs in an unbuffered system. Memory device selection must be qualified by a memory read or write to prevent spurious selection. Some Intel static RAM devices have an Output Enable for this purpose, such as the 2142 (1k x 4). This part was designed to be specifically used with a microprocessor bus. For other standard static RAMs, the chip selects must be qualified by RD, WR or ALE to prevent random selection.

#### DYNAMIC RAM INTERFACE

An earlier Intel Application Report (APR-1) extensively covered dynamic RAM interface with different types of memory and refresh in the MCS-80 system. This dynamic RAM section was taken from the most memory intensive example in APR-1, the 2116, modified to be compatible with the 8085A bus. These minor modifications are such that an 8080 system can be converted without much trouble. Before discussion of this section, however, a strong word of advice is in order. At about the same time this Application Note is published, Intel will be sampling an 8202 dynamic RAM refresh controller which does all dynamic RAM interfacing (except the data bus) and refreshing in one packaged component. It is highly recommended that the reader investigate this before using the attached schematic. Reading this section will still be useful in terms of understanding the 8085A bus.

This section uses the APR-1 2116 (multiplexed address 16K) example modified for the 2117-4 dynamic RAM. These devices have some differences from the 2116. One is that the output is not latched and is 3-stated during a write operation. This allows a user to tie both the data in and data out pins together at the device and at the data buffers, saving board traces. The 2117 also have hidden refresh capabilities where if  $\overline{CAS}$  is held low,  $\overline{RAS}$  can be toggled to refresh the device.

The schematic shown in Figure 7 is aimed at a high performance, relatively inexpensive solution (disregarding the 8202). Refresh circuitry is not shown, but can be implemented in a variety of ways. This will be discussed later in an upcoming section. In this refresh section, code for a simple, very low cost refresh controller that requires no special hardware, other than an 8155 timer, is presented.

For system timing, a 4x clock is used to obtain the resolution necessary to provide the clocks for the multiplexed address 2117's. Other solutions are possible with delay lines, one shots, etc., but are relatively expensive and don't provide for a nice baud rate source for any peripherals that may be in the system as does this 4x clock. Another approach can use the clock edges from the 8085A CLKOUT to interface to dynamic RAM. To facilitate this type of approach, Clock related timing parameters are listed later in this note.

To aid in understanding the operation of this circuit, the explanation is broken into a discussion of the main signal paths. 2117-4 Spec compatibility with the 8085A will be discussed in detail in the dynamic RAM section of the Memory Compatibility section.

#### Addresses

The lower 14 addresses (A0-A13) are used to select one of the 16,384 8-bit bytes in each 16K byte data bank. The lower 8 of these 14 addresses (A0-A7) flow through an 8212 and are latched by ALE, effectively demultiplexing the address/data bus. These lower 8 addresses with the next 6 (A8-A13) enter the 3242 multiplexer/refresh controller. The Row Enable of the 3242 controls which half of the addresses are presented to the dynamic RAM memory. Looking at the row enable on the 3242, it is seen that the row and column addresses are swapped with respect to convention. The higher order ad-

**Figure 7. 8085A-2117 Dynamic RAM Interface.**

addresses are used as row addresses and the lower order addresses are used as column addresses. This does not create problems because this is invisible to the CPU. Refreshing is done properly as the 3242 controls the addressing for this. The upper two address lines ( $A_{14}$ - $A_{15}$ ) are decoded to qualify one of the four  $\overline{RAS}$  (Row Address Strobe) lines to select one of the four 16K byte data banks of memory.

#### Cycle Requests

Cycle requests are generated from several sources; ALE automatically initiates a request when S1 indicates that there is a read taking place (flip-flop C), WR during write cycles (D) and refresh delayed (Q output of refresh flipflop (B)) when there is a refresh. ALE is used to start a read (qualified by S1) to provide ample time for access from the memories. This cycle request signal (A) immediately creates a  $\overline{RAS}$  and starts a timing chain (74S174 shift register (E)) to generate the remaining signals. Synchronization between this cycle request pulse and the 4x clock is accomplished by the first D flip-flop in the 'S174 shift register (timing chain).

#### $\overline{RAS}/\overline{CAS}$

When  $\overline{RAS}$  is enabled by a cycle request, it is qualified with either a refresh request (all  $\overline{RAS}$ 's turn on) or the decoded upper two bits of the address bus. A careful reader may question whether address is valid prior to  $\overline{RAS}$  being enabled. This question can be answered by noting that the 8212 passes the address through before the falling edge of ALE latches it.  $T_{AL}^{\dagger}$  (115 ns for 320 ns 8085A processor cycle), which is the time from address to the falling edge of ALE, gives ample time for addresses to be valid at the 3242 outputs before  $\overline{RAS}$  is valid.  $\overline{RAS}$  is extended past the clearing of the cycle request flip-flop by ORing this enabling signal with a tap from the D flip-flop shift register.

$\overline{CAS}$  (Column Address Strobe) is produced between 123 and 164 ns after  $\overline{RAS}$ , depending upon when the first D flip-flop in the shift register synchronizes with the cycle request signal (C). Since this is greater than the specified maximum delay from  $\overline{RAS}$  to  $\overline{CAS}$ , this memory system is  $\overline{CAS}$  access limited and  $\overline{RAS}$  access no longer has any meaning. The  $\overline{CAS}$  tap can't move up one D flip-flop to provide more time for memory access as this would not provide sufficient data set up time with respect to  $\overline{CAS}$  during a write.

#### Data

The data path to the 2117s is through two sets of buffers to account for memory being off board. To determine bus timing it is helpful to know that Write data is not guaranteed to be valid from the 8085A until 40 ns after the leading edge of the write control signal. On account of this and the delay times for the buffers it is necessary to delay the cycle request on a write until the WR signal goes low. The solution shown still does not require wait states. An inhibit memory signal is also involved. This is useful when using memory address space overlap such as the case with bootstrap ROM (which would be necessary in this system if a full 64K of dynamic RAM is used).

#### Refresh

Dynamic RAMs are generally refreshed in two different modes; burst (i.e., all at once every 2 ms) and distributed (one row every (2 ms/number of rows) period of time). The schematic shown provides for a distributed refresh where refresh requests are applied to the Hold request input of the 8085A (not shown). This signal needs to occur at least once

every 15  $\mu$ sec ((2ms/128 rows to be refreshed) - HOLD to HLDA delay) and can be generated through a baud rate timing chain, Intel 3222, one shots or other similar devices. Another approach to refresh could qualify the refresh cycles with program fetch cycles (use status lines). If program memory is in static RAM or ROM and the dynamic RAM bus can be isolated, refresh cycles can be performed with no overhead. Instead of using the HOLD feature of the 8085A, refresh can be hidden in the program fetch and decode. Further considerations for refresh include proper handling of resets and excessive hold times from other peripherals to be certain the memory is being refreshed adequately.

Some applications don't require high CPU efficiency and require a very inexpensive method to refresh their dynamic RAM. Since writing, reading or performing special refresh cycles all refresh a particular row, why not do "dummy" reads to refresh? To use this technique memory must be mapped on a one to one correspondence with the address space. This will allow the programmer to read one byte in each physical row in the 2117s, thereby refreshing that row. A simple software routine can be devised to refresh 16K bytes of RAM. If more dynamic RAM than this is desired it can be accomplished by specially enabling all the desired  $\overline{RAS}$  signals via an 8085A output port. First let's analyze how many CPU cycles are available in the 2ms period:

$$2\text{ms}/(320 \text{ ns/cycle}) = 6,250 \text{ cycles} \\ \text{for } 8085A @ 3.125 \text{ MHz}$$

$$2\text{ms}/(200 \text{ ns/cycle}) = 10,000 \text{ cycles} \\ \text{for } 8085A-2 @ 5.0 \text{ MHz}$$

If there is a convenient component that can count 8085A cycles (8085A CLKOUT) and interrupt the 8085A, you're home free. An example of such a device is the 8155 in the MCS-85 family. On the 8155 one can use the  $\overline{TO}$  (timer out) pin to interrupt the CPU everytime a refresh needs to be performed and an interrupt service routine could dummy read 128 consecutive locations and return to CPU operation. (128 reads are necessary to completely refresh the full 16K bytes of 2117 memory.) The highest priority interrupt should be used for this to insure that refresh occurs. Figure 8 is an example program to perform this burst dummy read refresh. This routine basically uses 64 pops of the stack, each reading two consecutive locations in the memory. Note that this routine destroys the contents of registers B, C and D in the 8085A. The user may want to save these registers in the routine before performing the software refresh. If memory space is more valuable than CPU efficiency, the POPs can be performed in a loop instead of a string, saving additional memory.

This routine requires 690 cycles which is about 11% of the available 8085A CPU cycles, or 7% of the available 8085A-2 cycles. If this is acceptable and there is a counter available, you can't find a cheaper way to do refresh. Note that as processor speeds become faster, this overhead becomes proportionately less and more attractive as an alternative. Again, as with any refresh routine, reset and excessive holds must be dealt with to guarantee proper refresh.

<sup>†</sup>Note that  $T_{AL}$  now only applies to the high order address byte.  $T_{ALL}$ , for the lower address byte equals 90 ns. This was done to allow for additional  $T_{RAE}$  time for data float.

```

MVI A, A4H    INTERRUPT CPU AT T0 (TIMER OUT)
OUT TIMER LSBYTE
MVI A, C0H    START COUNTER, PLACE C0 IN 8155 STATUS REG.
OUT TIMER COMMAND
Program

```

AT RST 7.5	RETURN ADDRESS	CALL RFRS	(REFRESH SERVICE)
TOTAL	#CYCLES		
CYCLES			
	10	RFRS: LXI HL, 0	SAVE STACK POINTER IN HL
	10	DAD SP	
30	10	LXI SP, 0080	32K - 48K REFRESH
	10	POP BC	
	10	POP BC	REFRESH, DUMMY READ
			64 TIMES
640	6	SPHL	RESTORE STACK POINTER
	4	EI	ENABLE INTERRUPTS
20	10	RET	RETURN
690	TOTAL CYCLES	(round up to 700)	

\*6,250 available cycles - 700 to do refresh. Counter should count 5550 = 15A4H for 8085A; for 8085A-2 must count 10,000-700 = 9300 = 2454H. To set counter to automatic reload, most significant bits in timer of 8155 must be set to 1. Therefore, for 8085A use D5A4H and for 8085A-2 use E454H.

Figure 8. Software Refresh.

#### DMA (Direct Memory Access)

DMA is becoming more common in the microcomputer system for many applications. Some examples include the 8271 floppy disk controller and refreshing a CRT via an 8275 CRT Controller. It is always helpful to reduce the overhead of the DMA (as DMA can tie up the system bus) whenever possible. In many applications, where program memory is resident in ROM or PROM, DMA cycles can be performed in coincidence with op code fetch. This will make them invisible to the CPU as described for Refresh in the Refresh section of the 2117 dynamic RAM example.

In the dynamic RAM system, Refresh requests can be made on the DMA controller via the DRQ lines, with the 8257 in a rotating priority mode to insure refreshing is done. Another technique would be to devise an arbiter for DMA and refresh requests at the processor hold input. With this technique the

designer must not allow DMA to monopolize the bus when refresh is needed.

The standard technique for interfacing the 8085A processor to the 8257 DMA controller is shown in the MCS-85 User's Manual and is reproduced in Figure 9. This configuration is set up to interface with standard memories or peripherals, i.e., ones that don't share their data bus with addresses, not the MCS-85 family components (8155, 8355, 8755A etc.). DMA is unlikely with these MCS-85 components as they are intended for minimum system applications. If the system has both MCS-85 and standard addressed components, and DMA is used for the standard addressed components, ALE must be or'ed with ADSTB from the 8257. This is necessary to deselect the MCS-85 components from the bus. Due to the latching feature of the MCS-85 components, bus contention may result if this is not done and DMA tries to use the bus.

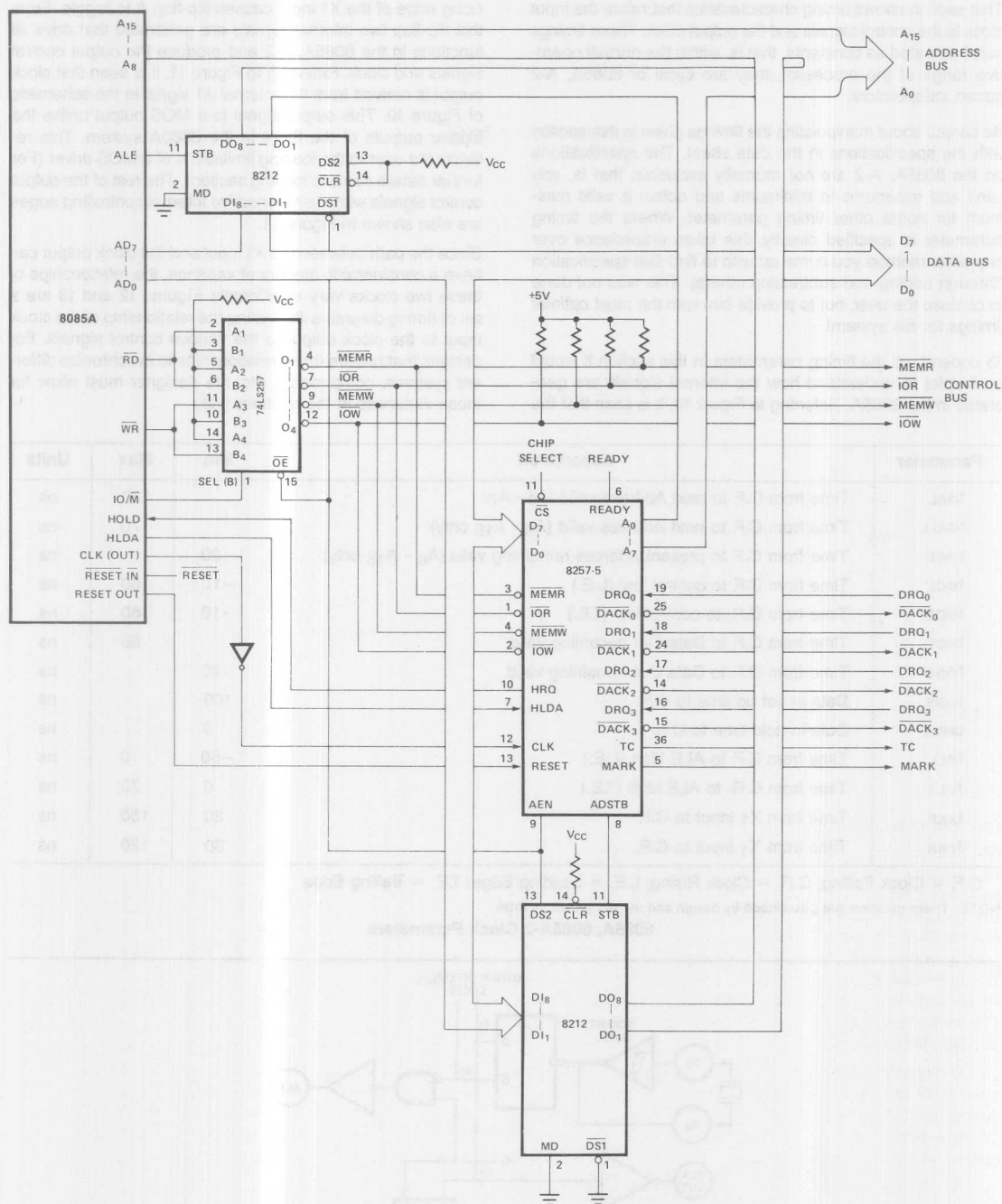


Figure 9. Detailed System Interface Schematic.

## SYSTEM TIMINGS

### 8085A CLK-IN vs. CLK-OUT vs. Control Timings

This section shows timing characteristics that relate the input clock to the control signals and the output clock. These timings can be treated as constants, that is, within the normal operative range of the processor, they are cycle or 8085A, A-2 speed independent.

Be careful about manipulating the timings given in this section with the specifications in the data sheet. The specifications on the 8085A, A-2 are *not* mutually exclusive; that is, you can't add minimums to minimums and obtain a valid minimum for some other timing parameter. Where the timing parameter is specified directly, this takes precedence over any other method you come up with to find that specification (through adding and subtracting others). This was not done to confuse the user, but to provide him with the most optimal timings for his system!

To understand the timing parameters in this section it would be helpful to understand how the internal signals are generated in the 8085A. Referring to Figure 10, it is seen that the

rising edge of the X1 input causes flip-flop A to toggle. From this flip-flop two internal signals are generated that drive all functions in the 8085A, A-2 and produce the output control signals and clock. Referring to Figure 11, it is seen that clock output is derived from the internal  $\phi_1$  signal in the schematic of Figure 10. This output signal is a MOS output unlike the bipolar outputs of the 8224 in the 8080A system. This restricts the user to the loading limitations of a MOS driver (For further details see bus loading section). The rest of the output control signals with their respective internal controlling edges are also shown in Figure 11.

Since the path between the X1 input and the clock output can have a considerable amount of variance, the relationships of these two clocks vary significantly. Figures 12 and 13 are a set of timing diagrams illustrating the relationship of the clock input to the clock output to the various control signals. For designs that require these relationships to synchronize different systems, components, etc; the designer must allow for these variances in the relationships.

Parameter	Description	Min	Max	Units
$t\phi_{AL}$	Time from C.F. to next Address valid ( $A_0 - A_7$ )		130	ns
$t\phi_{ALU}$	Time from C.F. to next Address valid ( $A_8 - A_{15}$ only)		70	ns
$t\phi_{AT}$	Time from C.F. to present Address remaining valid ( $A_8 - A_{15}$ only)	-20		ns
$t\phi_{CL}$	Time from C.F. to control low (L.E.)	-10	60	ns
$t\phi_{CT}$	Time from C.R. to control low (T.E.)	-10	60	ns
$t\phi_{DL}$	Time from C.F. to Data Out becoming valid		65	ns
$t\phi_{DT}$	Time from C.F. to Data Out remaining valid	-20		ns
$t\phi_{DS}$	Data-in set up time to C.R.	100		ns
$t\phi_{DH}$	Data-in hold time to C.R.	0		ns
$t\phi_{LL}$	Time from C.F. to ALE high (L.E.)	-60	0	ns
$t\phi_{LT}$	Time from C.R. to ALE high (T.E.)	0	70	ns
$t_{XKF}$	Time from $X_1$ input to C.F.	30	150	ns
$t_{XKR}$	Time from $X_1$ input to C.R.	30	120	ns

C.F. = Clock Falling, C.R. = Clock Rising, L.E. = Leading Edge, T.E. = Trailing Edge

NOTE: These numbers are guaranteed by design and are not tested by Intel.

### 8085A, 8085A-2 Clock Parameters

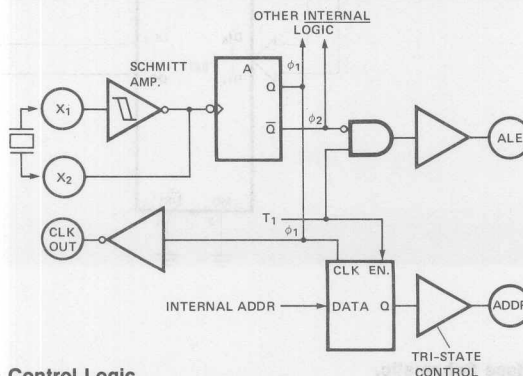


Figure 10. Clock and Sample Control Logic.

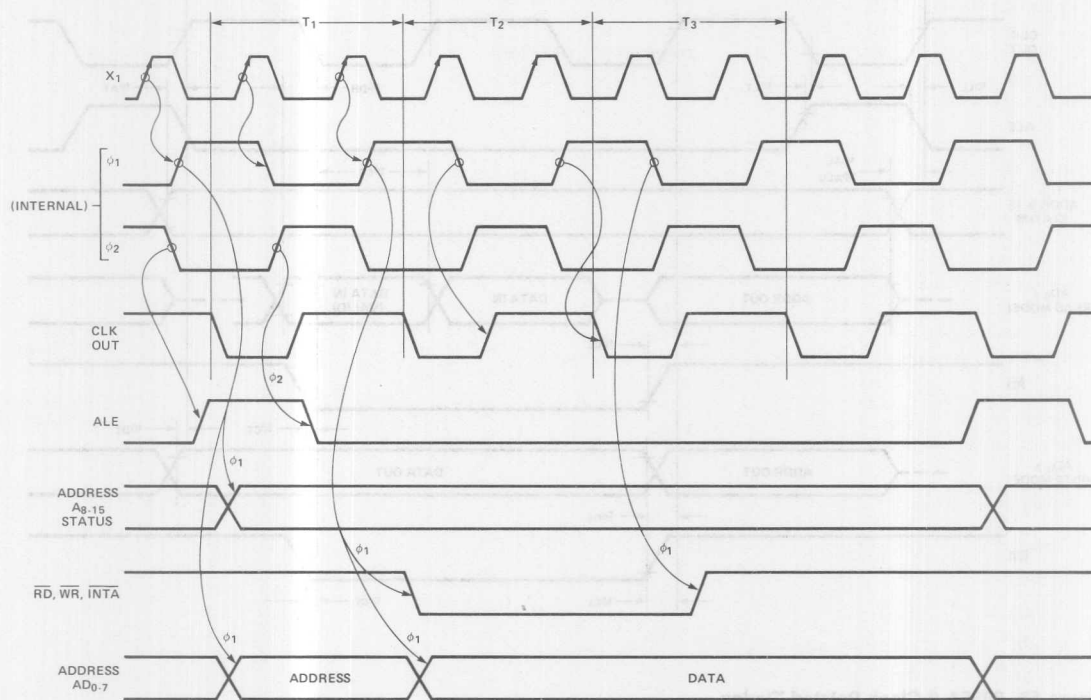


Figure 11. Clock In (X1) to Output Relationship.

—Clock (output) register. The 8085A-2 requires the following crystal speed: 10 to 15 MHz.

8085A-2 is a 16-bit data bus system. It can be used with 8085A-2 or 8085A-2. It can be used with 8085A-2 or 8085A-2. It can be used with 8085A-2 or 8085A-2.

8085A-2 is a 16-bit data bus system. It can be used with 8085A-2 or 8085A-2. It can be used with 8085A-2 or 8085A-2. It can be used with 8085A-2 or 8085A-2.

8085A-2 is a 16-bit data bus system. It can be used with 8085A-2 or 8085A-2. It can be used with 8085A-2 or 8085A-2. It can be used with 8085A-2 or 8085A-2.

8085A-2 is a 16-bit data bus system. It can be used with 8085A-2 or 8085A-2. It can be used with 8085A-2 or 8085A-2. It can be used with 8085A-2 or 8085A-2.

Table 1. 8085A-2

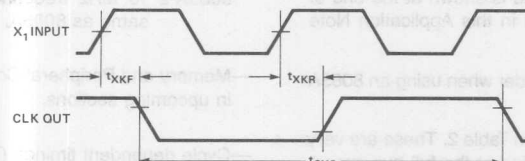


Figure 12. 8085A-2 Clock In/Clock Out Timing.

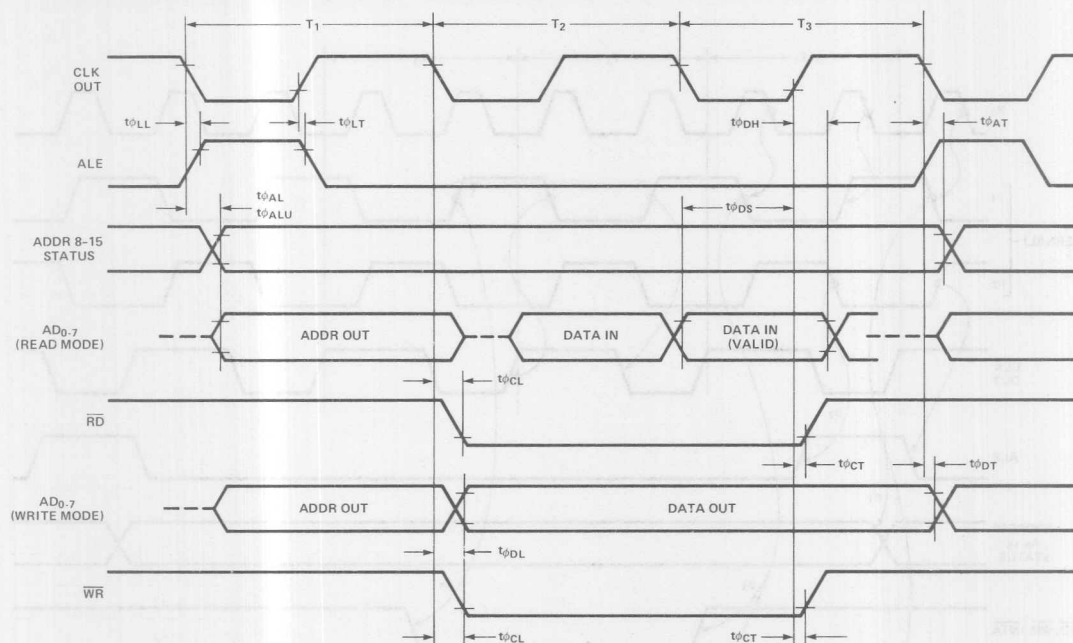


Figure 13. 8085A-2 Clock Related Timing.

### 3.125 vs. 5 MHz Considerations

The 8085A (with maximum internal clock frequency of 3.125 MHz) and 8085A-2 (5 MHz) have some differences in their bus operation. There are two sets of peripherals that can be used with both the 8085A and A-2. There are the dedicated peripherals in the MCS-85 family that directly interface with the 8085A, A-2 bus and the standard MCS-80 peripherals that Intel also provides. The standard peripherals that are denoted 825X-5 (also the 8251A and 827X peripherals) are peripherals that can be used with an 8085A or 8085A-2. In the 8085A-2 system a wait state is required for proper I/O operation, but even with this wait state system speed is still 30% higher than the 8085A without wait states. An example wait state generator for this purpose is shown at the end of the peripheral compatibility section in this Application Note (Fig. 16).

The main timing differences to consider when using an 8085A vs. an A-2 are listed in Table 1.

Cycle dependent timings are listed in Table 2. These are very useful when the user is not operating at the full bus speed. Remember that each 8085A, A-2 device divides its clock input frequency by 2. Therefore, a 10 MHz crystal will produce a 200ns output cycle (denoted as T in the cycle dependent timings). A timing diagram showing the relationships of the timing parameters given in Table 2 can be found on the data sheets.

—Clock (crystal) requirements The 8085A, A-2 requires the following crystal specifications to run at top bus speed:

8085A 6.25 MHz frequency, parallel resonant, fundamental, 10 mwatt drive level,  $R_S < 75$  ohms,  $CL = 20-35$  pf, and  $CS < 7$  pf.

8085A-2 10 MHz frequency, all other specifications the same as 8085A.

—Memory and Peripheral Compatibility - Discussed in detail in upcoming sections.

—Cycle dependent timings (Table 2)

Table 1. 8085A vs. 8085A-2.

**PRELIMINARY**  
 Notice: This is not a final specification. Some  
 parameter limits are subject to change.

Parameter	3.125 MHz (8085A)			5 MHz (8085A-2)		
	(ns)		Cycle Dependencies	(ns)		Cycle Dependencies
tcyc	320	2000		200	2000	
t1	80		1/2T-80	40		1/2T-70
t2	120		1/2T-40	70		1/2T-50
tr		30			30	
tf		30			30	
tAL	115		1/2T-45	50		1/2T-50
tLA	100		1/2T-60	50		1/2T-50
tLL	140		1/2T-20	80		1/2T-20
tLCK	100		1/2T-60	50		1/2T-50
tLC	130		1/2T-30	60		1/2T-40
tAFR		0			0	
tAD		575	(5/2+N)T-225		350	(5/2+N)T-150
tRD		300	(3/2+N)T-180		150	(3/2+N)T-150
tRDH	0			0		
tRAE	150		1/2T-10	90		1/2T-10
tCA	120		1/2T-40	60		1/2T-40
tDW	420		(3/2+N)T-60	230		(3/2+N)T-70
tWD	100		1/2T-60	60		1/2T-40
tCC	400		(3/2+N)T-80	230		(3/2+N)T-70
tCL	50		1/2T-110	25		1/2T-75
tARY		220	3/2T-260		100	3/2T-200
tRYS	110			100		
tRYH	0			0		
tHACK	110		1/2T-50	40		
tHABE		210	1/2T+50		150	1/2T+50
tRV	400		3/2T-80	220		3/2T-80
tAC	270		T-50	115		T-85
tHDS	170			120		
tHDH	0			0		
tINS	360		1/2T+200	150		1/2T+50
tINH	0			0		
tLDR		460	2T-180		270	4/2T-130

Where T = tcyc and N = the number of wait states that are incorporated.  
 All mathematical operations in Table 2 are performed from left to right, except where qualified with parenthesis.

Table 2. 8085A and 8085A-2 Cycle Dependencies.

### Memory Device Compatibility

## Determining What Memory to Select For Your Application

When developing a system which will use sufficient memory to require buffering (see the capacitive loading section to determine when it is needed), it is important to understand how to select the slowest, lowest cost memory and still be compatible with the bus timings with minimum wait states. A generalized procedure has been developed in the following section for determining the memory access needed for different applications and the number of wait states required (if any). In general the amount of time available for accessing the memory can be obtained from the following formula: Available memory access = 8085A access time (from control signal of interest) - Buffering/Decoding delay (to and from memory)

The three main "control" signals of interest which determine memory access are that of  $t_{RD}$  (read to valid data in),  $t_{AD}$  (valid address to valid data in) and  $t_{LDR}$  (address latch enable to valid data in). When dealing with different types of memories, one or more of these signals becomes important. Even though memory access compatibility is probably one of the most important parameters to consider, as this is directly reflected in the price of the memory, it is not the only parameter that is important. Some of the other major timing considerations are as follows:

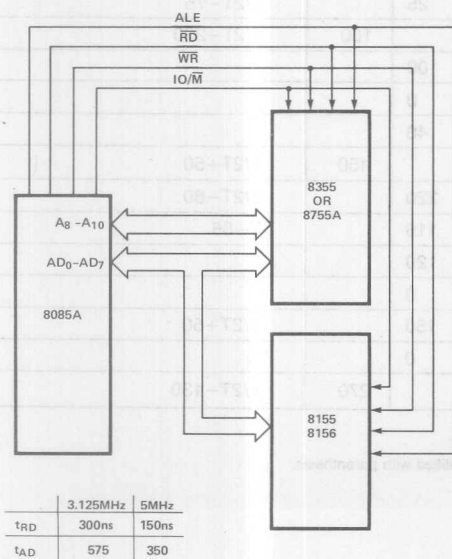
**WRITE ENABLE** - Is the write enable signal sufficiently long to guarantee a write?

Is data set up properly with respect to this write to be compatible with the memory's requirements?

Is data held long enough?

**DATA FLOAT** - Does your system have sufficient margin to prevent bus contention?

(i.e., Does the memory let go of the data bus in time for the processor to use it? Remember that the 8085A shares its Data Bus with the lower 8 addresses.)



**Figure 14. Minimum System.**

We will first go through the minimum system which can be represented by the dedicated set of components Intel has developed for the 8085A (Fig. 14). The two timing specs were taken from the data catalog for  $t_{RD}$  and  $t_{AD}$  ( $t_{LDR}$  is irrelevant here). Looking at the 8155/6 and 8355/8755A, a comparison can be made for the access times:

	8085A (3.125 MHz)	8155/6	8355/8755A
t <sub>RD</sub>	300 (max)	170 (max)	170 (max)
t <sub>AD</sub>	575 (max)	400 (max)	400/450 (max)

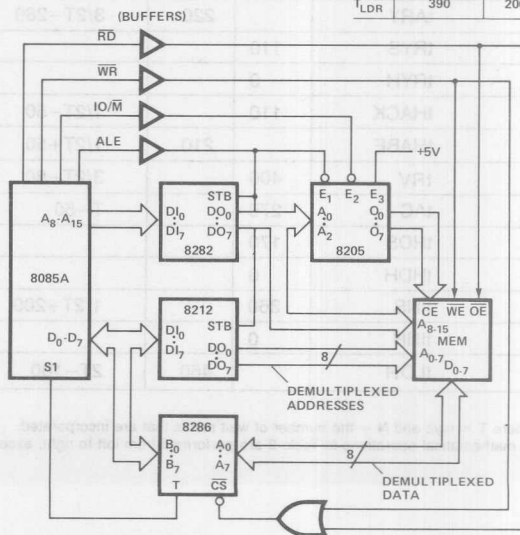
This shows that there is plenty of bus margin for the 3.125 MHz minimum application of the 8085A. Access time for the processor can be interpreted as the time from when the control signal is presented on the bus to the time when the processor will expect the data to be valid so it can sample it. Conversely, memory access times show the amount of time that will elapse between when it is told to present its information to when it actually does it. As long as the memory access spec is less than the processor access spec (minus appropriate buffering delays) the memory is access time compatible.

In more complicated systems where one level of data, address and control buffering is required (such as the case when there are many signal paths and device loading on one card), the delays of the latches and bidirectional drivers must be taken into consideration.

First consider a ROM, EPROM or static RAM configuration as shown in Figure 15. Using the generalized available memory access formula,  $t_{AD}$ ,  $t_{RD}$  and  $t_{LDR}$  for the memory can be determined using the data sheet timing delays for the buffers.

† This configuration is compatible with the edge enabled ROMs discussed earlier.

	3.125MHz	5MHz
$T_{RD}$	250	100
$T_{AD}$	500	275
$T_{LDR}$	390	200



**Figure 15. Medium Buffered System.**

$$\begin{aligned}
t_{AD} \text{ MEMORY} &= t_{AD8085A} - (8282 + 8205 \text{ delay}) - (8286 \text{ delay}) + \text{transitional gain due to buffering}^* \\
&= t_{AD85} - (T_{IVOV} + t_{--}) - (T_{IVOV}) + t_{CAPB}^* \\
&= (5/2 + N)T - 225 - 55 - 35 + 15 \\
&= (5/2 + N)T - 300 \quad (\text{for } 8085A) \\
&= (5/2 + N)T - 225 \quad (\text{for } 8085A-2)
\end{aligned}$$

where N = number of wait states and T = cycle time,

For minimum 8085A timing 500ns =  $t_{AD}$  memory

8085A-2 timing 275ns =  $t_{AD}$  memory

The 8085A timing parameter  $t_{AL}$  was not taken into consideration as the 8282 transfers information directly through without concern of the address latch enable.  $t_{RD}$  can be obtained in a similar manner.

The read signal  $\overline{RD}$  goes through a buffer before it reaches the memory. This must be taken into consideration when calculating effective  $t_{RD}$  for the memory.

$$\begin{aligned}
t_{RD} \text{ MEMORY} &= t_{RD} 8085A - (\text{buffer delay}) - (8286 \text{ delay}) + \text{transitional gain due to buffering} \\
&= t_{RD} 85 - (\text{delay}) - (T_{IVOV}) + t_{CAPB} \\
&= (3/2 + N)T - 180 - 30 - 35 + 15 \\
&= (3/2 + N)T - 230 \quad (\text{for } 8085A) \\
&= (3/2 + N)T - 200 \text{ ns} \quad (\text{for } 8085A-2)
\end{aligned}$$

\* $t_{CAPB}$  is additional time thrown back in for improvement in signal transitions. This is because buffering the signals reduces the capacitive loading considerably. The data sheet gives timings for maximum capacitive loading. Characterization has shown change in delay versus capacitive loading as .12 ns/pf min (under 20 pF loading) and .24 ns/pF max (under 150 pF loading). To take into consideration the effects of this loading two parameters are defined:

$t_{CAPA}$  - delay for a signal to leave the old logic level

$t_{CAPB}$  - delay for a signal to complete the transition from the old to new logic level

where  $t_{CAPA} = 1/2 t_{CAPB}$

	MIN	MAX
$t_{CAPA}$	7 ns	15 ns
$t_{CAPB}$	15 ns	30 ns

In the memory compatibility calculations  $t_{CAPB}$  min is added on as spec sheet values assume 150 pF loading and this system is not worst case, i.e., it has buffering that reduces this loading to approximately 20 pf. Since the CAP = 130 pF and change in delay versus capacitance is 1/2 ns/pF min,  $t_{CAPB} \text{ MIN} = (.1 \text{ ns/pF}) 130 \text{ pF} = \text{approx. } 15 \text{ ns}$ .

For minimum 8085A timing 250ns =  $t_{RD}$  memory

8085A-2 timing 100ns =  $t_{RD}$  memory

With the new Intel Edge Enabled ROMs (2332, 2364) 8085A ALE can be conveniently used for generating CE.

Therefore for  $t_{LDR}$ :

$$\begin{aligned}
t_{LDR} \text{ MEMORY} &= t_{LDR} 8085 - (\text{buffer delay}) - (8205) \\
&\quad - (8286) + t_{CAPB} \\
&= t_{LDR} - (\text{delay}) - (t_{--}) - (T_{IVOV}) + t_{CAPB} \\
&= 2T - 180 - 30 - 20 - 35 + 15 \\
&= 2T - 250 \text{ for } 8085A \\
&= 2T - 200 \text{ for } 8085A-2
\end{aligned}$$

For minimum 8085 timing = 390ns

8085A-2 timing = 200ns

To obtain memory access parameters for a multiscard system (which would have buffering at both ends of the system bus), it is a simple matter of subtracting off the additional buffering delays.

With these timings a memory compatibility table can be developed from the data sheets (Table 3). With most of these memories it is relatively straightforward to determine the controlling signal used to select and enable the device. To illustrate this, listed below are the controlling signals of interest for the different memories as they are used in a typical configuration:

		Relevant Control Signal
RAM 2114	Address access	- $t_{AD} \text{ MEM}$
	Chip select access	- $t_{LDR} \text{ MEM}^{**}$
2142	Address access	- $t_{AD} \text{ MEM}$
	Chip select access	- $t_{LDR} \text{ MEM}^{**}$
	Output enable	- $t_{RD} \text{ MEM}$
ROM 2332, 2364	Address access	- $t_{AD} \text{ MEM}$
	Chip enables	- $t_{LDR} \text{ MEM}$
	Output enable	- $t_{RD} \text{ MEM}$

\*\*Chip selects for these static RAMs need not be qualified with ALE. If 2114 or 2142 chip selects are generated directly from the address lines, the relevant timing is  $t_{AD} \text{ MEM}$ .

	3.125 MHz	5MHz
<b>MINIMUM SYSTEM:</b>		
STATIC RAM	8155/8156, (256x8)	8155-2/8156-2
	8185 (1Kx8)	8185-2
ROM/EPROM	8355 (2Kx8)	8355-2
	8755A (2Kx8)	8755A-2
<b>BUFFERED SYSTEM:</b>		
STATIC RAM	2114 (1Kx4)	2114-2
	2142 (1Kx4)	2142-2
ROM/EPROM	2332 (4Kx8)	*
	2364 (8Kx8)	
	2732 (4Kx8)	
	2716-2 (2Kx8)	2716-2**

\*Contact Intel for high performance EPROM/ROM Family.

\*\*With 1 wait state.

**Table 3. 8085A, A-2 Memory Compatibility.**

In general,  $t_{AD}$  MEM and  $t_{LDR}$  MEM are the parameters needed for chip enabling, selection and address access times, and probably are the most important considerations when determining which memory device to use. When there is an output enable,  $t_{RD}$  MEM is also used. All relevant access times must be met by the resulting system configuration to be compatible.

This note will not attempt to generalize a procedure that deals with the interface to dynamic RAM, but the 2117 example shown earlier is described below. In the dynamic RAM system, many variables come into play upon which the memory access is dependent. Among these are refresh controllers, decoding, whether or not the system is designed for minimum hardware or maximum performance, and consideration for nonmultiplexed vs. multiplexed address dynamic RAMs.

For the Intel® 2107C, which has nonmultiplexed addresses,  $t_{AD}$  is the important parameter as it generates the chip selects and chip enables. However, with a multiplexed address part, things are different and both a  $\overline{RAS}$  and  $\overline{CAS}$  access time must be considered. Note that since  $\overline{RAS}$  is applied before  $\overline{CAS}$ ,  $\overline{RAS}$  access time is effective only while the  $\overline{CAS}$  signal stays within the specified  $\overline{RAS}$  to  $\overline{CAS}$  delay time. If it is not possible to do this,  $\overline{CAS}$  access becomes the limiting factor for memory selection. Don't be misled by the  $\overline{RAS}$  to  $\overline{CAS}$  maximum delay ( $t_{RCD}$ :  $\overline{RAS}$  to  $\overline{CAS}$  delay time) spec'd on dynamic RAM data sheets! This maximum only applies to guarantee  $\overline{RAS}$  access.

For a specific example the following shows how the speed versions were selected for previous 2117 dynamic RAM interface.

$\overline{RAS}$  path (from ALE) approximate delay

5 gates 7 ns ea  
1 Flip Flop 15 ns  
(return path) 2 8216s 25 ns ea

$\overline{CAS}$  path (from ALE) approximate delay

3 gates 7 ns ea  
1 Flip Flop 15 ns  
4 D Flip Flops 41 ns ea

$t_{ACCESS}$  AVAILABLE FOR  $\overline{RAS}$  =

$$t_{LDR} - 5(7) - 15 - 2(25) = 360 \text{ ns}$$

$t_{ACCESS}$  AVAILABLE FOR  $\overline{CAS}$  =

$$t_{LDR} - 3(7) - 15 - 4(41) - 2(25) = 210 \text{ ns}$$

Since  $\overline{RAS}$  available time -  $\overline{CAS}$  available time is greater than the spec value for  $\overline{RAS}$  to  $\overline{CAS}$  delay on all 2117 specs,  $\overline{CAS}$  access becomes the limiting factor. A  $\overline{CAS}$  access of 165ns of the 2117-4 is well within the time available.

To verify the other 2117 specs such that there is certainty that this system will play, a comparison can be made of the timing specs in the 2117 data sheet to the timings that result in the circuit configuration in Figure 8. When looking at the following timing comparisons, remember that the read cycle is initiated by the falling edge of ALE (Address Latch Enable) and the write from the falling edge of  $\overline{WR}$  (Write). For descriptions of the parameters in Table 4, please refer to a 2117-4 data sheet. Delay assumptions used are shown in Table 5.

#### TAKEN FROM 2117-4 DATA SHEET

READ CYCLE	MIN	MAX
t <sub>RAC</sub>		250 ns
t <sub>CAC</sub>		165 ns
t <sub>REF</sub>		2 ms
t <sub>RP</sub>	150 ns	
t <sub>CPN</sub>	25 ns	
t <sub>CRP</sub>	-20 ns	
t <sub>RCD</sub>	35 ns	65 ns
t <sub>RS</sub>	165 ns	
t <sub>CS</sub>	250 ns	
t <sub>AS</sub>	0 ns	
t <sub>RAH</sub>	35 ns	
t <sub>ASC</sub>	-10 ns	
t <sub>CAH</sub>	75 ns	
t <sub>AR</sub>	160 ns	
t <sub>off</sub>	70 ns	
t <sub>RC</sub>	410 ns	
t <sub>RAS</sub>	250 ns	
t <sub>CAS</sub>	165 ns	

\*There are two parameters that the processor "sees". One is memory access, which has already been covered. The other is when the memory will let go of the bus. To show compatibility here, the following analysis is done:

2117 t<sub>OFF</sub> 70 ns max  
8085A t<sub>RAE</sub> 150 ns min

Therefore compatible as  $\overline{WR}$  is used to deselect the 8216's.

#### DYNAMIC RAM CONFIGURATION

	MIN	MAX
	Doesn't apply	
	210 ns	
	Not Shown	
	279 ns	
	472 ns	
	193 ns	
	Outside spec, CAS access limited	
	177 ns	
	300 ns	
	55 ns	
	82 ns	
	-4 ns	
	205 ns	
	410 ns	
	See Below*	
	720 ns	
	307 ns	
	198 ns	

Table 4: Figure 7 Bus Compatibility Analysis.

TAKEN FROM 2117-4 DATA SHEET			DYNAMIC RAM CONFIGURATION	
WRITE CYCLE	MIN	MAX	MIN	MAX
tRC	410 ns		720 ns	
tRAS	250 ns		307 ns	
tCAS	165 ns		198 ns	
tWCS	-20 ns		34 ns	
tWCH	75 ns		164 ns	
tWCR	160 ns		287 ns	
tWP	75 ns		205 ns	
tRWL	100 ns		205 ns	
tCWL	100 ns		205 ns	
tDS	0 ns		23 ns **	**Data is not valid from the 8085A until 40 ns after WR falls.
tDH	75 ns		Data held until next cycle	
tDHR	160 ns		Data held until next cycle	

Table 4. (Cont'd)

The numbers in Table 4 were obtained by using the following delay assumptions (Table 5) and very conservative techniques of obtaining minimum 8085A timings. Where no direct specification applied, minimum specs were added assuming 0 ns for any rise or fall times. This is more conservative than necessary. Another approach can be made from the clock related timings discussed in an earlier section.

	DELAY	
	MIN	MAX
Gates	0 ns	7 ns
Flip Flops	0 ns	15 ns
8216s	0 ns	30 ns
D flip flop (Timing Chain)	41 ns	41 ns
	(Min Ons for synchronization D FF)	
3242	0 ns	25 ns
8212	0 ns	30 ns

Table 5. Delay Assumptions.

An exhaustive approach as Table 4 will more than pay itself back in terms of debugging the circuit. However, while this analysis may be helpful in understanding an existing circuit, it won't help as much in creating a new one. A general procedure for designing with memories is itemized below:

1. Determine how much processor time is available for memory access. Access from addresses is the most important parameter.
2. Determine how much buffering will be used (both to and from the memory) and how much delay there will be due to decode or qualifications in the circuit (in the memory design in Fig. 8,  $\overline{WR}$  qualifies a write). Subtract these resulting delays from step 1 to get an effective access for the memory. If multiplexed address RAM is used go to 3, if not go to 4.
3. Determine how the  $\overline{RAS}$  and  $\overline{CAS}$  timings will be generated, be it one shots, delay lines, shift registers, etc. Adjust memory access available for the method chosen.
4. Select a memory that meets this criterion.
5. Design the system to meet all the specified parameters of the memory and verify.

Steps 1, 2 and 4 have been done for you in the Memory Compatibility Table for ROM, EPROM and Static RAM memories in a medium and minimum system. *Remember* - for dynamic RAM, Intel will soon be providing an 8202, a refresh, dynamic RAM controller that generates all  $\overline{RAS}$ ,  $\overline{CAS}$  control signals for a 64 kByte memory (made of 2117s).

#### Peripheral Compatibility - 3.125 and 5 MHz

Intel supports its processors with many LSI peripheral components that do a wide range of functions to simplify circuit design. The 8085A compatible peripherals have been denoted the "-5" notation to show compatibility. The "-5" notation also signifies that these devices are compatible with the 8085A-2 with one wait state interjected. This wait state is produced by taking the ready line low at the proper time as shown in Figure 16.

A list of these peripherals is shown in Table 6 with corresponding relevant specifications to illustrate 8085A-2 compatibility. The analysis for determining the resulting timings is similar to the analysis in the previous memory compatibility section.

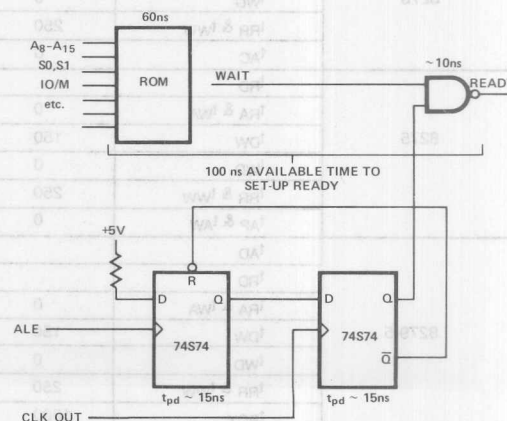


Figure 16. 8085A-2 Wait State Generator.

Part No.	AC. Parameter	Min. (ns)	Max. (ns)	8085A-2 AC. Parameter	Margin vs. -2 Spec. (ns)
8251A	t <sub>RD</sub>		200	t <sub>RD</sub>	*150
	t <sub>RA</sub> & t <sub>WA</sub>	0		t <sub>CA</sub>	60
	t <sub>DW</sub>	150		t <sub>DW</sub>	80
	t <sub>WD</sub>	0		t <sub>WD</sub>	60
	t <sub>RR</sub> & t <sub>WW</sub>	250		t <sub>CC</sub>	*180
	t <sub>AR</sub> & t <sub>AW</sub>	0		t <sub>AC</sub>	
8253-5	t <sub>RD</sub>		200	t <sub>RD</sub>	*150
	t <sub>RA</sub>	5		t <sub>CA</sub>	55
	t <sub>WA</sub>	30		t <sub>CA</sub>	60
	t <sub>DW</sub>	250		t <sub>DW</sub>	*180
	t <sub>WD</sub>	30		t <sub>WD</sub>	30
	t <sub>RR</sub> & t <sub>WW</sub>	300		t <sub>CC</sub>	*130
	t <sub>RV</sub>	1000		t <sub>RV</sub>	**
	t <sub>AR</sub> & t <sub>AW</sub>	50		t <sub>AC</sub>	65
8255A-5	t <sub>RD</sub>		200	t <sub>RD</sub>	*150
	t <sub>RA</sub>	0		t <sub>CA</sub>	60
	t <sub>WA</sub>	20		t <sub>CA</sub>	40
	t <sub>DW</sub>	100		t <sub>DW</sub>	130
	t <sub>WD</sub>	30		t <sub>WD</sub>	30
	t <sub>RR</sub> & t <sub>WW</sub>	300		t <sub>CC</sub>	*130
	t <sub>RV</sub>	850		t <sub>RV</sub>	**
	t <sub>AR</sub> & t <sub>AW</sub>	0		t <sub>AC</sub>	115
8257-5	t <sub>RD</sub>		200	t <sub>RD</sub>	*150
	t <sub>RA</sub> & t <sub>WA</sub>	0		t <sub>CA</sub>	60
	t <sub>DW</sub>	200		t <sub>DW</sub>	30
	t <sub>WD</sub>	0		t <sub>WD</sub>	60
	t <sub>RR</sub>	250		t <sub>CC</sub>	*180
	t <sub>WW</sub>	200		t <sub>CC</sub>	30
	t <sub>AR</sub>	0		t <sub>AC</sub>	115
	t <sub>AW</sub>	20		t <sub>AC</sub>	95
8271 & 8273	t <sub>AD</sub>		200	t <sub>AD</sub>	*350
	t <sub>RD</sub>		150	t <sub>RD</sub>	*200
	t <sub>CA</sub>	0		t <sub>CA</sub>	60
	t <sub>DW</sub>	150		t <sub>DW</sub>	80
	t <sub>WD</sub>	0		t <sub>WD</sub>	80
	t <sub>RR</sub> & t <sub>WW</sub>	250		t <sub>CC</sub>	*180
	t <sub>AC</sub>	0		t <sub>AC</sub>	115
	t <sub>RD</sub>		200	t <sub>RD</sub>	*150
8275	t <sub>RA</sub> & t <sub>WA</sub>	0		t <sub>CA</sub>	60
	t <sub>DW</sub>	150		t <sub>DW</sub>	80
	t <sub>WD</sub>	0		t <sub>WD</sub>	60
	t <sub>RR</sub> & t <sub>WW</sub>	250		t <sub>CC</sub>	*180
	t <sub>AP</sub> & t <sub>AW</sub>	0		t <sub>AC</sub>	115
	t <sub>AD</sub>		250	t <sub>AD</sub>	300
8279-5	t <sub>RD</sub>		150	t <sub>RD</sub>	200
	t <sub>RA</sub> & t <sub>WA</sub>	0		t <sub>CA</sub>	60
	t <sub>DW</sub>	150		t <sub>DW</sub>	80
	t <sub>WD</sub>	0		t <sub>WD</sub>	60
	t <sub>RR</sub> & t <sub>WW</sub>	250		t <sub>CC</sub>	*180
	t <sub>RCY</sub>	1000		t <sub>RV</sub>	**
	t <sub>AP</sub> & t <sub>AW</sub>	0		t <sub>AC</sub>	115

Table 6. Peripherals vs. 8085A-2.

\*With 1 "Wait State"

\*\*Must allow for in Software

Taking note of asterisked margins shown on the comparison sheet:  $t_{AD}$ ,  $t_{RD}$ ,  $t_{RR}$  and  $t_{DW}$ , it is seen that they are all taken care of by introducing a wait state. The double asterisked margins deal with the  $t_{QV}$  spec on the 8255A-5, 8253-5 and 8279-5 peripherals.  $t_{QV}$  is the time from the rising edge of  $WR$  or  $RD$  to the next falling edge. To allow sufficient time for this spec it is necessary to delay the commands sent to these three peripherals. Enough dead time must occur to make up for the entire negative portion of the margin (for example: 790ns in the 8253-5 medium system). Since in the 8085A-2 every machine cycle is at least 200ns long, 4 machine cycles are sufficient time to allow peripheral control signal recovery ( $t_{QV}$ ).

One may notice that all of the 8085A instructions take at least 4 T-states (providing a minimum of 800ns) giving ample time to meet this requirement, just by programming one instruction in between every command sent to the peripheral. I/O mapped I/O, which results in using the Input, Output instructions has this delay time built in when moving the data to be transferred into the accumulator. With memory mapped I/O, any instruction that accesses memory for data will provide the time necessary to not violate  $t_{QV}$  as a second fetch is performed.

#### Bus - loading considerations - decoupling

For the cost conscious designer it is always helpful to know when buffering is needed and when it is not. How much can I load the 8085A output pins down? To answer this it is helpful to first list the DC requirements of the common types of logic loading and compare this to the capabilities of the 8085A.

	Maximum High-Level Input Current	Maximum Low-Level Input Current
TTL (single load)	40 $\mu$ A	1.6mA
Schottky or HTTL	40 $\mu$ A	2.0mA
MOS	10 $\mu$ A	10 $\mu$ A
LSTTL (single load)	20 $\mu$ A	400 $\mu$ A

The 8085A is capable of an IOL of 2mA (low) and IOH of - 400 $\mu$ A. With this spec it is easy to come up with the possible combinations of D.C. loading that the designer can use without buffering:

LOADS	8085A, A-2 limiting factor (level)
1 TTL + 1 LSTTL	LOW
1 TTL + 36 MOS*	HIGH
1 SCHOTTKY or 1 HTTL	LOW
40 MOS (various combinations possible)*	HIGH
5 LS TTL	LOW

\* Exceeds capacitive loading limit, to be discussed

If a user exceeds these DC loading limitations he must buffer that particular signal. Another factor that the designer must consider is the capacitive load that is seen by the 8085A outputs, which may very well be excessive even if DC loading is not. One may note that even though the 8085A can handle a DC load of 40 MOS devices or 36 MOS + 1 TTL, their collective input capacitances exceed the 150 pF max spec.

The timing specs of the 8085A are guaranteed as long as the 150 pF maximum loading is not exceeded, which includes the wires, components and parasitics. If the user exceeds this value and wants to guarantee his system timing he must either derate the system timings or use buffering.

What if you choose to ignore this limit and say you can live with the performance degradation? First the timing performance is not all that would degrade, a user must be willing to give up some reliability of his components (All MOS devices have this restraint). This is caused by the excessive switching currents that are needed for this extra loading capacitance. If reliability is not an important consideration, the user can load up to 300 pF on the 8085A bus, but the following correction factors must be used to adjust the timings:

for 150 pF < 300 pF add .13 ns/pF

conversely if less than 150 pF:

for 25 < CL < 150 pF you can subtract .1ns/pF.

What happens after 300 pF? If the user exceeds this, the noise levels become excessive and problems will result. How much is too much noise? 350 mvolts zero to peak. Prudent designers will always buffer when noise approaches this level, especially in the case of going from one board to another.

The above takes into consideration the actual specification considerations of when to buffer, but there are also transmission line and noise effects that must be considered. When working with dynamic RAMs small (20-30 ohm) resistors are commonly put in series in the address lines to help match impedance levels and reduce reflections. Note that this resistor should be chosen such that it does not severely degrade the voltage levels of the signal. Long parallel board traces with signals that could adversely affect each other should also be avoided to prevent cross talk problems.

By-passing is very important to prevent intermittent problems which often plague the board designer. Large bulk capacitors should be used at strategic locations on the board to prevent power supply droop. This becomes a major factor when there are many devices that can turn on at once and produce a considerable drain from the power supply (such as burst refresh in dynamic RAM).

To help smooth out the current spikes that naturally occur when devices turn on and off, it is recommended to liberally use small capacitors such as the monolithic and other ceramic capacitors which have low inherent inductance. Attached in the 2117 data sheet is a suggested layout of capacitors to effectively bypass the supply lines to ensure proper system operation. Cutting corners here will often times turn around and bite you.

Proper layout is an important consideration. Power supply lines should be well gridded to supply sufficient current to all areas of the board. A strong ground layout is advised to offset noise problems. Remember if the ground plane moves up in voltage because of excessive charge dumping in a particular area, the supply will drift up correspondingly. Sensing low levels often becomes an intermittent problem when proper ground is not provided.

Following is an application example that illustrates the use of the interrupt and SOD pins on the 8085A, software for a block search routine, and the procedure for using and reading the 8155 counter. It is a simple application showing the use of the small but powerful 3-chip MCS-85 system as a temperature sensor (SDK-85 board used). This example can be modified to be an accurate industrial temperature controller, for several locations if desired.

The basic operation behind this application is a monostable multivibrator having its timing pulse duration controlled by a thermistor. The counter in the 8155 converts this timing pulse to a decimal count that is software mapped into a temperature and displayed in degrees C in the address field of the display in the SDK-85 Kit. For the purpose of keeping the software relatively simple, many approximations were incorporated into the code.

### Detailed Hardware

The basic SDK kit was used for the initial hardware. This Kit provides for everything necessary to develop and debug a program through the use of the SDK-85 monitor, keyboard and display board. The kit provides for 256 bytes of RAM resident in the 8155 and 2K bytes of ROM or EPROM where the SDK-85 monitor is placed. (See the Intel SDK-85 User's Manual for copy of monitor software code.)

Figure 17 is a schematic of the SDK-85 Kit with only one 8155 and 8355. There is no buffering in this system as all compo-

ponents are on the same board and far below the maximum component loading. A monostable multivibrator (74121) is also shown with a thermistor connected to RE/CE.

The SOD output pin from the 8085A is used for the purpose of starting the monostable multivibrator in generating its temperature controlled timing pulse. This pulse is created by the RC time constant provided for by the thermistor acting as a variable resistor and a .1 $\mu$ F capacitor to put the timing pulse in the desired timing range.

The inverted output of the monostable multivibrator (one shot) has been directly connected to the RST 6.5 pin on the 8085A. Since this pin is high level sensitive, it is necessary to disable interrupts in the program until after the pulse from the one shot goes low.

The hardware addressing in the configuration shown allows for several code spaces that could be used. The RST and TRAP interrupt lines on the 8085A also have hardware start addresses but many of these are altered by the SDK monitor. Table 7 should be useful in understanding the addresses used in the software that follows. Each memory/ I/O component in the basic SDK-85 system is enabled by a signal coming from the 8205 address decoder. Since no expansion chips are used, output enables 00 (8355 monitor ROM), 03 (8279 Keyboard) and 04 (8155 RAM) were the only ones needed. Additional memory and/or I/O could have been incorporated using other output enables from the 8205.

Memory/ I/O Device	Function	Output from 8205	code space
8155	RAM space	04	2000 - 20FF (20 - 20FF are reserved for monitor RAM locations)
8355	ROM space	00	0000 - 07FF
8279	Keyboard/display controller	03	1800 - 1FFF

stack pointer

Since the monitor uses locations 10C8 through 20FF, the stack pointer must be initialized to 20C8 or less.

	8085A jump address	Usage	monitor mapped address
trap	24H	T0 of 8155	0157
RST 5.5	2CH	8279 interrupt	028E
RST 6.5	34H	oneshot interrupt	20CE
RST 7.5	3CH	vector interrupt	
I/O ports address	Function		
00	Monitor ROM Port A (8355)		
01	Monitor ROM Port B (8355)		
02	Monitor ROM Port A (8355) Data direction register		
03	Monitor ROM Port B (8355) Data direction register		
20	Basic command/status register		
21	Basic RAM Port A		
22	Basic RAM Port B		
23	Basic RAM Port C		
24	Basic RAM LOW order byte of timer count		
25	Basic RAM HIGH order byte of timer count		

Table 7. Addressing

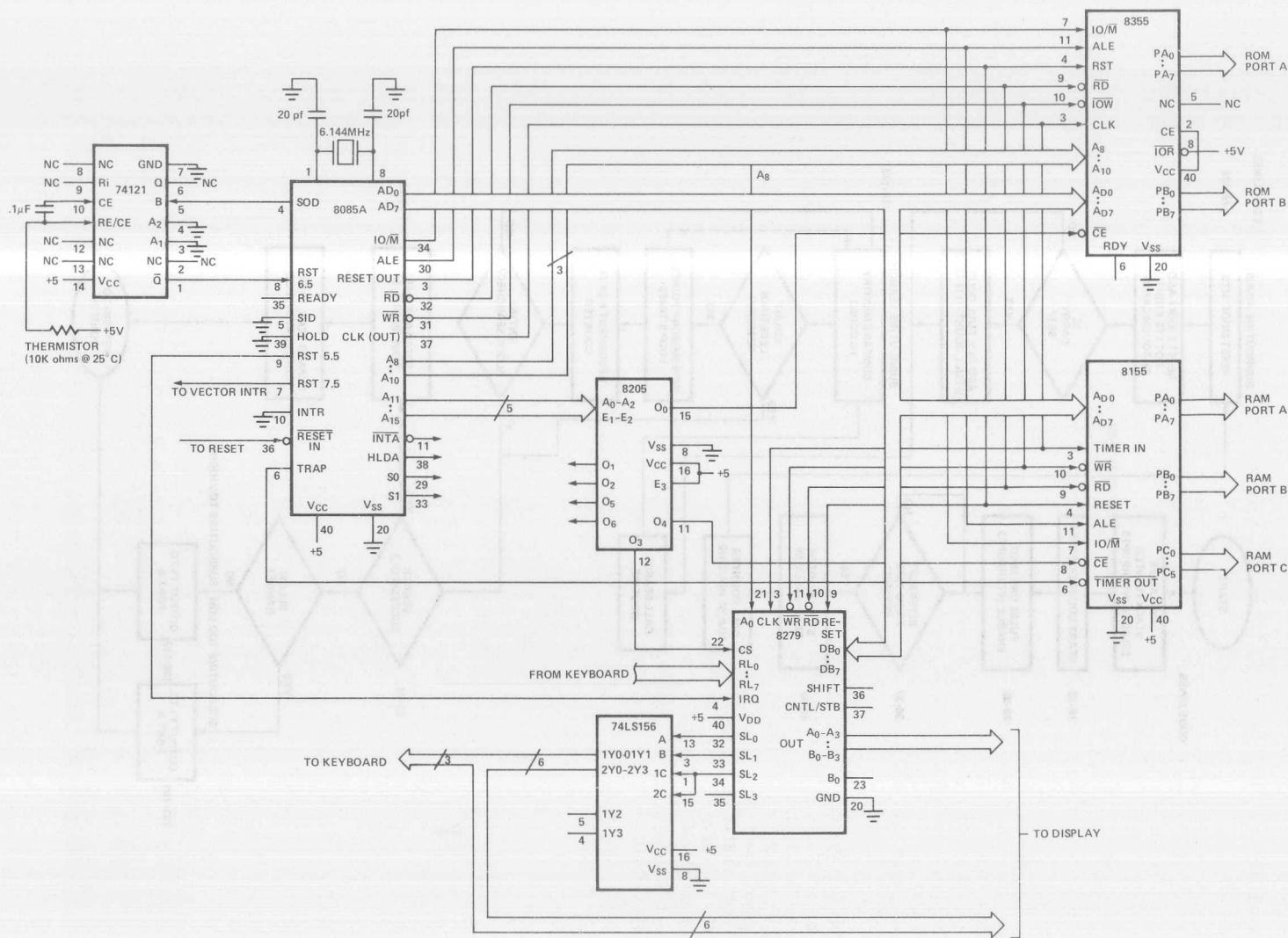


Figure 17. Detailed SDK-85 Kit with Temperature Sensor.

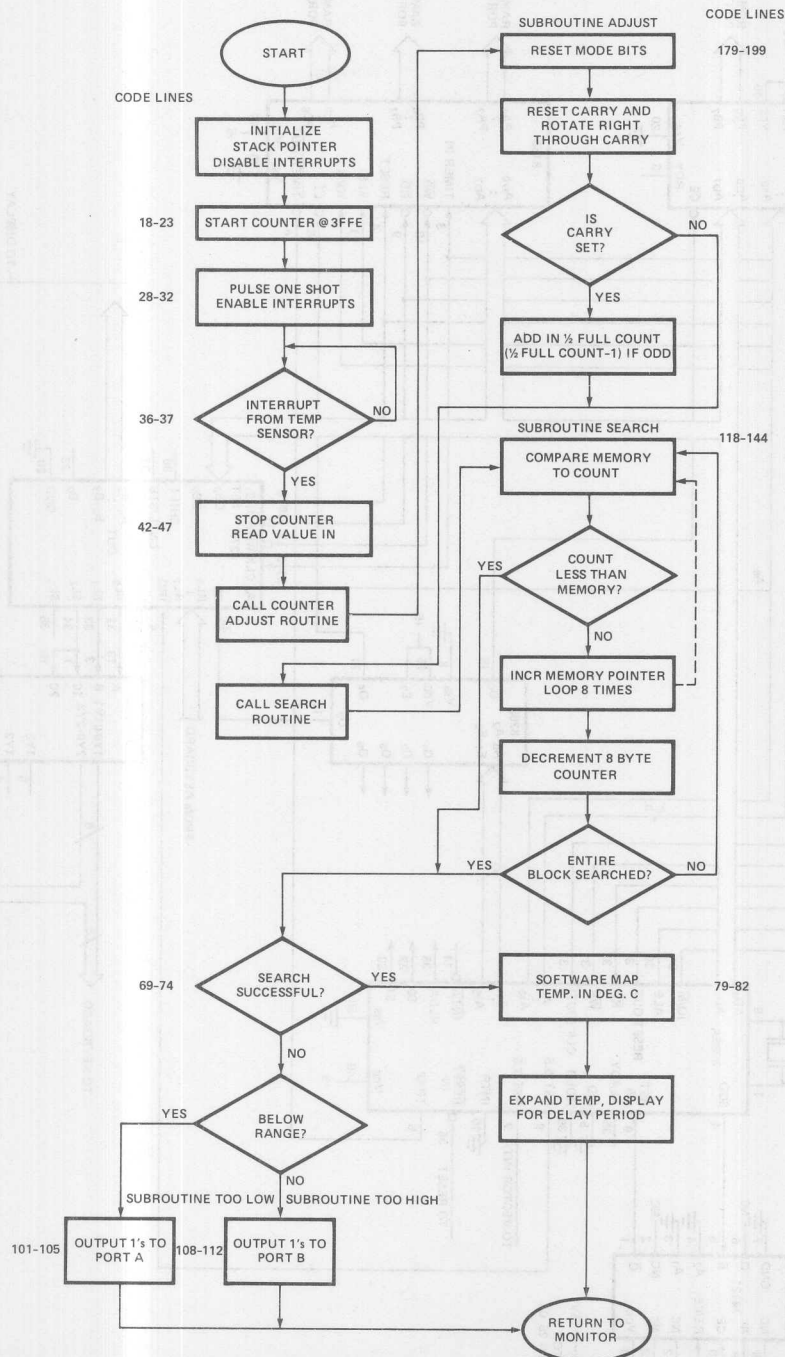


Figure 18. Temperature Sensor Flow Diagram.

## Software

The software (at end of section) for this application illustrates several features of the 8085A, such as the programming of the SOD line, interrupts and 8155 counter. Additionally, an example of a block search routine is illustrated.

Figure 18 is a flow diagram of the program. It has been cross referenced with program lines to the actual software for the reader's convenience. Following through the flow diagram it is seen that the interrupts are disabled in the beginning as the one shot is outputting a high level on its Q output and interrupt pin 6.5 is high level sensitive. However, this high level will not be recognized until the level goes low and then high again. If the user would prefer a positive pulse interrupt the 8085A a dual one shot can be used with one triggering the other, or just a simple inverter. Starting and loading the counter is as described in the 8155 data sheet with the Port addresses being given in the previous Table (7). Code lines 18-23 represent placing the counter in the counter mode (single terminal count pulse at the end of count) and starting the count, having the count clocked by the 8085A clock out pin. Reading the counter is not as straight forward and will be approached shortly. Code lines 28-32 are representative of programming the SOD line to output a pulse. This pin is intended for serial I/O interfaces such as a teletype, but as seen in this application, it can also be used as a single I/O port.

After the pulse is presented to the one shot, the interrupts enabled, the processor idles (lines 36, 37; Halt could have just as easily been used) until interrupted. Through the design of this application it was known that the down counter would never reach terminal count, as it is only being used as a pulse to digital count converter.

To read in the count value it is best that the counter is first stopped. The least and most significant bytes of the count length register in the 8155 are read using the same port addresses as was used during loading the counter, as seen in code lines 42-47. If one looks at this value and knows how many pulses occurred, he would come to the conclusion that there is a gross discrepancy! The reason for this is that the counter in the 8155/6 was designed to make its square wave function generation easy and when used in the counter mode, it counts by two's. For this application (where length of time is mapped into a temperature) and other similar event timing applications it is imperative to have an intelligible count returned from the 8155.

The counter in the 8155 is essentially a count down by 2 counter. After it counts down by 2 the initial value loaded by the user, it reloads the initial count (initial count -1 if odd) and counts down by 2 again until terminal count is reached. When reading the counter, the least significant bit of the counter does not represent the least significant bit of the count, but which half of the countdown operation you are in. If this bit equals 1, the 8155/6 counter is counting down by 2 in the first half, and if it is zero you are in the second half of the operation. Because of this method of down counting there are two restrictions placed on its use:

1. The user can not use the initial value of 1 to detect only one pulse.
2. The user can not discern (through reading the counter) whether exactly one or two pulses on the timer input pin has occurred if he loaded in an initial odd count (does not apply to even). After three pulses the user can determine exactly how many pulses occurred. Note that this restriction only applies to reading the counter, the  $\overline{TO}$  pin pulses correctly after the correct number of pulses regardless of what is read from the counter.

The first pulse to the 8155/6 counter (high level sensitive) loads the count length register, which says that the counter is not readable until a pulse occurs. If the user tries to read before a pulse is provided he will read a previous or old value. Now what is done with the value read?

Good question. An adjustment routine to convert this value read to an actual count can be summarized as follows:

1. Read in 16 bit count length register.
2. Reset the upper two bits (mode bits).
3. Reset carry and rotate right all 16 bits through carry.
4. If carry is set add 1/2 of full original count (1/2 (full count -1) if full count is odd).

In the software for this application is a general purpose routine to do this; lines 179-199. To call this routine it is assumed that the lower order byte of the counter is in register C, higher order byte in register B and full original count is in HL. Contents of H, L, B and C are destroyed returning actual count in BC register pair. To obtain the number of pulses that occurred, subtract this number from full original count and add 1.

Converting this remaining count to an actual temperature can be done by various methods but it was chosen to do a software map through the use of a block search routine. Table 8 presents approximations of what the remaining count should be for each temperature. To keep the software simple it was only necessary to compare the most significant byte to a list to find the appropriate temperature. This search routine is set up to find a "less than" match, incrementing the HL register as a pointer when a compare is made. The code for this search routine is in lines 118-144 and is optimized to be a fast 8 byte block search. This search routine can be made to search for a match by replacing all return on carry with return on zero. The performance of this subroutine is as follows:

$$\text{Byte time} = (11 + (166/8) N) \text{ CC}/N = (11/N + 20.8) \text{ CC}$$

where: CC = microseconds per clock cycle  
N = total number of bytes searched  
Byte time = time per byte searched



For an example with  $N = 256$ ,  $CC = .32 \mu\text{sec}$  at 3.125 MHz; Byte time =  $6.7 \mu\text{sec}$ . A match search routine with minimum memory usage is given below:

Search	Cmp M	compare byte
	RZ	return if match
	INX H	else increment pointer
	DCR C	has the entire
	JNZ search	block been searched?
	STC	If so set no match flag
	RET	and return.

In this application, a user may want to have several temperature ranges which can be swapped in and out with a block move subroutine. Similar code can be developed for this as shown below for a 4 byte move group:

BLKMV	LXI H, 000H	clear HL
	DAD SP	move SP to HL
	SHLD SAVESP	save sP
	MOV H, B	move Block move
	MOV L, C	Source address
	SPHL	To SP
	XCHG	Move Block move
		address to HL

Loop	POP B	fetch four bytes from
	POP D	source store 1st byte
	MOV M, C	at destination
	INX H	
	MOV M, B	2nd
	INX H	
	MOV M, E	3rd
	INX H	
	MOV M, D	4th
	INX H	
	DCR A	check for end of
	JNZ Loop	Block move
	LHLD SAVESP	return old
	SPHL	SP
	RET	return

Once the count less than match is found in the application the HL register has 10 added to it which points it at the corresponding temperature (lines 79-82). This temperature is then displayed in the address field of the SDK 85 display using user available monitor routines. If the temperature is out of range the code detects it (lines 69-74) and outputs 1's on Port A or Port B if the temperature was too low or too high respectively (lines 101-105 "too low" and lines 108-112 "too high").

## APPLICATION EXAMPLE 2

### CRT INTERFACE

Most microprocessor systems require some sort of serial communications. This may be selected for reasons of economy (to reduce the number of interconnections required in a distributed system), or it may be necessary in order to communicate with such common peripherals as CRT's or teletypewriters.

These peripherals all use a standard convention for transmitting serial ASCII code. Each data byte is transmitted as a series of 10 or 11 bits. The uniform time per bit corresponds to the data transmission rate. For example, if the transmission rate is to be 2400 baud (2400 bits per second), each bit time must be  $1/2400 \text{ bps} = 416.7 \mu\text{sec/bit}$ . The standard 10-bit sequence consists of a logically zero "Start" bit, 8 data bits (least significant bit first), and one or more stop bits (logic 1). An 11-bit sequence with two stop bits is used for 110 baud TTY's. The logic one level continues until the start bit of the next byte to ensure that each 10-bit sequence is initiated with a one-to-zero transition. The 8 bits transferred might be raw binary data or alphanumeric characters using the standard ASCII code. In this case, the most significant bit — the last data bit transmitted — will depend on the parity convention being used. This sequence is illustrated for the ASCII "space" character in Figure 4.

The algorithm for receiving serial code involves sampling the incoming data at the middle of each bit time. The eight sampled values are shifted into a serial byte corresponding to the data originally transmitted. The one-to-zero transition at the beginning of each byte makes it possible to synchronize the sampling points relative to the start of each data sequence.

### Hardware Interface

In general, any serial communications system will require both hardware and software interfaces. Since the SOD line can drive only one TTL load, additional current and voltage buffering is required to be compatible with the RS-232C interface standard used by most peripherals. A schematic for achieving this buffering is shown in Figure 5. The MC1488 and MC1489 circuits interface positive logic TTL signals with the RS-232 high voltage inverted logic levels.

### Software Package

The software needed to drive the CRT interface is divided into three parts. All three use software timing and delay loops, with fixed and variable parameters. In conjunction, they are able to identify incoming signals at any rate from below 110 to over 9600 baud and respond at the same rate.

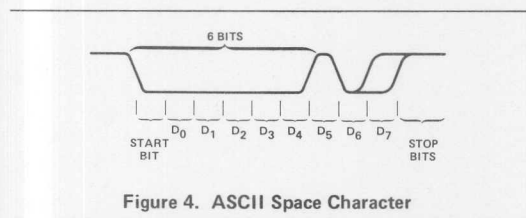


Figure 4. ASCII Space Character

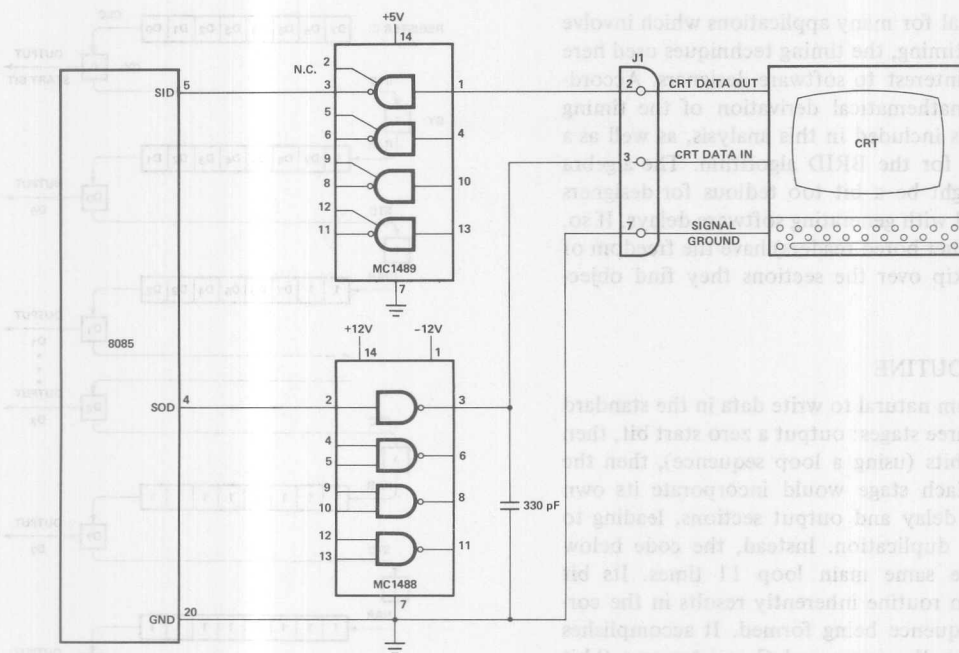


Figure 5. RS-232C Interface Schematic

Upon power-up or reset, or when the console baud rate is changed, the baud rate identification subroutine (BRID) is called. This routine waits until an ASCII space character (20H) is received from the console. (Any other character will result in a case of mistaken identification.) When a space character is received, two time parameters are computed which correspond to the bit time and one-half the bit time of the baud rate being used. These are stored as variables BITTIME and HALFBIT. To output a character to the console, the character code is placed in register C, and the subroutine COUT is called. This routine uses BITTIME as a parameter for the software delay loop which determines the baud rate. To accept a character from the keyboard, CIN is called. CIN returns after the next key is typed, with the corresponding character code in register C. CIN uses both parameters BITTIME and HALFBIT.

Since COUT and CIN use time parameters computed by BRID, they will function at a rate the same as that of the initial space character input. Because of the nature of the software, the rate does not depend on the CPU clock frequency. This

results in additional flexibility in the following respects:

1. The software does not need to be modified if the 8085 crystal frequency is changed or Wait states are added.
2. Since the time base is no longer critical, the quartz crystal could be replaced by a less expensive RC network, provided the frequency does not drift by more than a few percent during a session. Additional drift can be accommodated by periodically recalling the BRID routine.
3. Communication is possible at non-standard baud rates which relaxes the constraints on system peripherals.

It should be noted, though, that slowing down the CPU clock will decrease its throughput proportionately. In addition, it will degrade the maximum resolution of the delay loops, with the result that the highest baud rates may no longer be achievable.

A more detailed analysis of the CRT interface routines will be presented in the order of increasing complexity: COUT, CIN, and BRID. Since SID and

SOD are ideal for many applications which involve critical I/O timing, the timing techniques used here may be of interest to software designers. Accordingly, the mathematical derivation of the timing parameters is included in this analysis, as well as a justification for the BRID algorithm. The algebra involved might be a bit too tedious for designers unconcerned with generating software delays. If so, they (and other bored readers) have the freedom of choice to skip over the sections they find objectionable.

## OUTPUT ROUTINE

It would seem natural to write data in the standard format in three stages: output a zero start bit, then the 8 data bits (using a loop sequence), then the stop bits. Each stage would incorporate its own appropriate delay and output sections, leading to unnecessary duplication. Instead, the code below executes the same main loop 11 times. Its bit manipulation routine inherently results in the correct data sequence being formed. It accomplishes this by using the carry and C register as a 9-bit pseudo-circular shift register. Initially CY=0. The algorithm outputs CY, waits one bit time, sets CY=1, and then rotates the pseudo-register right one bit. This repeats for 11 cycles. On the tenth and all subsequent loops, the output bit will be a logical one, since that bit had been set nine loops earlier while in the CY (see Figure 6).

When COUT is called the registers to be used must be preserved and interrupts disabled so the timing loop will not be disrupted. Clear the CY in preparation for outputting the start bit, and set the loop counter for 11 bits (if 110 baud will never be used, the counter could be set to 10):

```
COUT:  PUSH  B
        PUSH  H
        DI
        XRA  A
        MVI  B, 11
```

Output of the contents of the CY:

```
CO1:   MVI  A, 90H    <7>
        RAR          <4>
        SIM          <4>
```

The numbers in brackets indicate how many machine cycles are required for each instruction. They will be referred to in the timing analysis section.

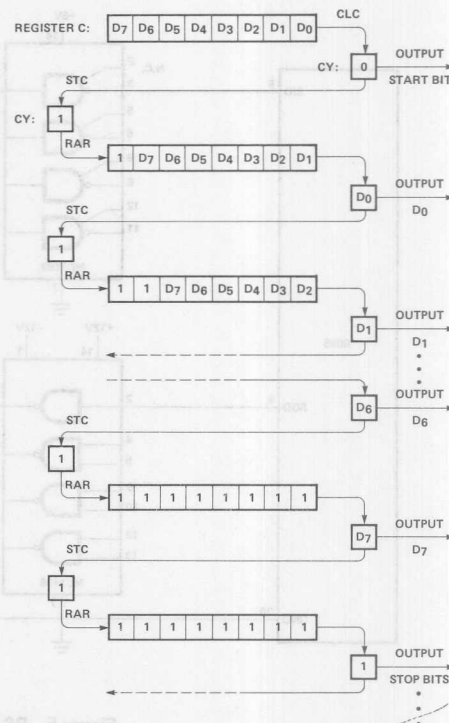


Figure 6. Data Serialization Algorithm

Get stuck in a loop for the appropriate time (don't worry for now how "BITTIME" is determined):

```
LHLD  BITTIME    <16>
CO2:  DCR  L      <6>
      JNZ  CO2    <6>
      DCR  H      <6>
      JNZ  CO2    <6>
```

Rotate the contents of register C right into the CY, while moving a one into the left end. Continue until all bits have been transmitted:

```
STC          <4>
MOV  A, C    <4>
RAR          <4>
MOV  C, A    <4>
DCR  B      <4>
JNZ  CO1    <10>
```

Restore processor status and return:



For a typical calculation, see Example 4.

#### EXAMPLE 4

To produce 2400 baud with the standard 6.144 MHz crystal:

$$2400 = \frac{(6.144 \times 10^6) \div 2}{83 + 14 \langle \text{HL} \rangle'}$$

$$14 \langle \text{HL} \rangle' = \left( \frac{6.144 \times 10^6 \div 2}{2400} \right) - 83$$

$$\langle \text{HL} \rangle' = \left[ \left( \frac{6.144 \times 10^6 \div 2}{2400} \right) - 83 \right] \div 14 = 85.5 \cong 86$$

$$\langle \text{HL} \rangle' = 86_{10} = 0056\text{H}$$

$$\langle \text{HL} \rangle = 0157\text{H} = \text{BITTIME}$$

To determine the true data rate this parameter will produce, substitute into equation (6):

$$\text{Data Rate} = \frac{6.144 \times 10^6 \div 2}{83 + 14(86)}$$

$$= 2387 \text{ baud, which is } 0.54\% \text{ slow.}$$

For 9600 baud, the same calculations will yield  $\langle \text{HL} \rangle' = 17$ , which is actually 0.3% slow; a sizzling 19200 baud or 38400 baud could each be generated to within 5% if  $\langle \text{HL} \rangle' = 6$  or 0! Table 1 presents the parameters for several standard baud rates.

Notice that the resolution of the delay algorithm — the difference between bit times resulting from parameters which differ by one — is 14 machine cycles. As a result, the true bit delay produced can always manage to be within  $\pm 2.3 \mu\text{sec}$  of the delay

desired. This guarantees that at rates up to 9600 baud, where each bit time is at least  $104 \mu\text{sec}$  wide, some value of BITTIME can be found which will be accurate to within 2.2%.

#### BAUD RATE IDENTIFICATION ROUTINE

The function of BRID is to compute the appropriate parameters BITTIME and HALFBIT. It accomplishes this by observing the data pattern received when the space bar is pressed on the console device. Since a space character has the ASCII code  $20\text{H} = 00100000\text{B}$ , the pattern represented back in Figure 4 is transmitted. Notice that the initial zero level is 6 bits wide. Suppose it could be determined that this corresponds to  $M$  machine cycles. Then one bit would correspond to  $(M \div 6)$  machine cycles. The reason for dividing down a space several bits long is so that any distortion caused by the signal rise and fall times, or any lack of precision in detecting the two transitions, will be reduced by a factor of six. Since the bit period of COUT and CIN is  $83 + 14 \langle \text{HL} \rangle'$ , BRID must generate a value  $\langle \text{HL} \rangle'$  such that:

$$M \div 6 = 83 + 14 \langle \text{HL} \rangle' \quad (7)$$

$$\langle \text{HL} \rangle' = \frac{(M \div 6) - 83}{14} \quad (8)$$

$$\langle \text{HL} \rangle' = \frac{M}{84} - 6 \text{ (approximately)} \quad (9)$$

This value can be determined by setting register pair HL to  $-6$ , then incrementing it once every 84 machine cycles during the period that the incom-

Table 1

DELAY PARAMETERS FOR STANDARD BAND RATES USING 6.144 MHz CRYSTAL

TARGET BAUD RATE	$\langle \text{HL} \rangle'_{10}$ (See Text)	$\langle \text{HL} \rangle'_{16}$ (See Text)	$\langle \text{HL} \rangle$ or BITTIME (See Text)	HALFBIT	ACTUAL BAUD RATE PRODUCED	% ERROR
110	1989	07C5	08C6	04E3	109.99	-0.006
150	1457	05B1	06B2	03D9	149.99	-0.005
300	726	02D6	03D7	026C	299.80	-0.068
600	360	0168	0269	01A5	599.65	-0.059
1200	177	00B1	01B2	0159	1199.5	-0.039
2400	86	0056	0157	012C	2386.9	-0.547
4800	40	0028	0129	0115	4777.6	-0.469
9600	17	0011	0112	0109	9570.1	-0.312
19200	6	0006	0107	0104	18395.2	-4.37

ing signal is zero. BITTIME is then obtained by individually incrementing registers H and L. To obtain HALFBIT, divide the value of (HL) determined above by two before incrementing each register.

In order to implement this algorithm, set HL to -6, verify that the incoming signal is a logic one, then wait for the start bit transition.

```
BRID: MVI A, 0C0H
      SIM
      LXI H, -6H
BR11: RIM
      ORA A
      JP BR11
BR12: RIM
      ORA A
      JM BR12
```

Increment register pair HL, then delay so that each cycle will require 84 machine cycles:

```
BR13: INX H
      MVI E, 04H
      DCR E
      JNZ BR14
      INX L
      JNZ BR14
```

Check if SID is still low. If so, repeat:

```
BR15: RIM
      ORA A
      JP BR15
```

Otherwise continue. Store HL temporarily for the HALFBIT calculation. Obtain and store BITTIME:

```
PUSH H
INR H
INR L
SHLD BITTIME
```

Restore HL, calculate HALFBIT, and return:

```
POP H
ORA A
MOV A, H
RAR
MOV H, A
RAR
MOV A, L
RAR
MOV L, A
INR H
INR L
SHLD HALFBIT
RET
```

The assembled listings for these subroutines, along with a simple test program, is presented in the Appendix.

There are many situations where data has to be transmitted through a non-ideal medium. To give three typical examples, a system with electrically isolated elements might require that signals be AC coupled, communications through an audio network (such as telephone or radio) are greatly bandwidth limited, and some applications (such as a distributed network in an industrial environment) must tolerate random electrical noise. Attempting to record data on a cheap cassette recorder (the one used for this note cost \$17.00) will reveal all of these shortcomings, plus one: The tape speed fluctuates significantly and varies as the batteries run down, hence the data rate is inconsistent.

The recording scheme used here makes very few demands on the transmission medium. It makes no attempt to transmit DC voltage levels. Instead, data is transmitted by a series of variable length tone bursts. The dominant frequency of the tone used can be selected to be within the passband of the particular medium. Data is transmitted with each bit composed of a tone burst followed by a pause. The first third of a bit period is always a tone burst, the middle third is either a tone burst continuous with the first or a pause corresponding to, respectively, a one or zero, and the final third is always a pause, as shown in Figure 7. Thus, data is distinguished by the burst/pause ratio.

### Hardware Design

These tone bursts are obtained from the 8085 SOD line, using analog signal conditioning to eliminate the DC component of the waveform. (This low frequency component is due to the single-ended nature of the SOD line: its deviations from ground are all positive, which unbalances the capacitive input stage of the recorder.) A suggested interface

circuit is shown in Figure 8, using one LM324 quad op amp and a few standard value discrete components which should be available in even a digital design laboratory. On playback, analog circuitry is again used to detect the presence of a tone burst. In Figure 8, A2 buffers the incoming signal, and A3 inverts it. The peaks of these two signals are transmitted through D1 or D2 and are filtered by an RC network. Comparator A4 then squares up the output and produces the logic signal read by the SID pin. Since the op amps are powered by the single 5-volt supply, a 2.0-volt reference level is obtained from a resistive voltage divider. The waveforms present at several points in the circuit are shown in Figure 9.

### Software

The algorithm for reading a data bit off the tape is simple and straightforward: If the tone burst is longer than the pause, the bit is a one. Otherwise, it is a zero. Since only the time ratio is considered, any variation in tape speed will not affect the data determination.

### VOLUME CONTROL

A question that arises with any audio cassette interface is how to set the volume control. (Recording level is usually determined internally.) When the playback level is correct, the logic signal output from A4 will have either a one-third or two-thirds duty cycle. This can be readily observed with an oscilloscope. In the field, an old-fashioned mechanical-type voltmeter could be connected to the A4 output, and the volume adjusted until the meter needle hovered somewhere between 1/3 and 2/3 the high level output voltage. With random data, the reading would be about 2 volts. There will be a fairly wide range of acceptable volume settings. (Since the quivering meter needle is being used here for inertial signal averaging, a digital voltmeter would not be very helpful in this application.)

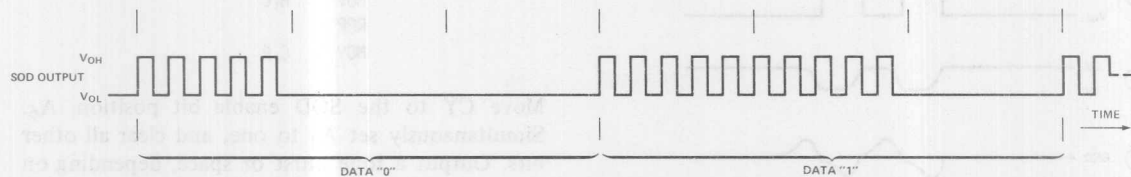
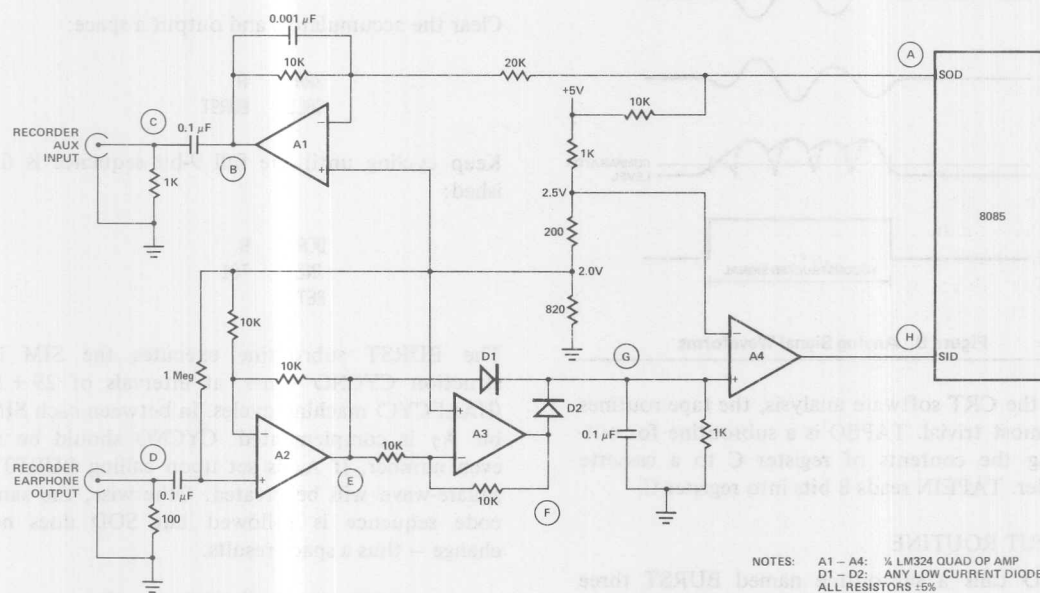


Figure 7. Tape Interface Data Recording Scheme



**Figure 8. One Chip Magnetic Tape Interface Schematic**

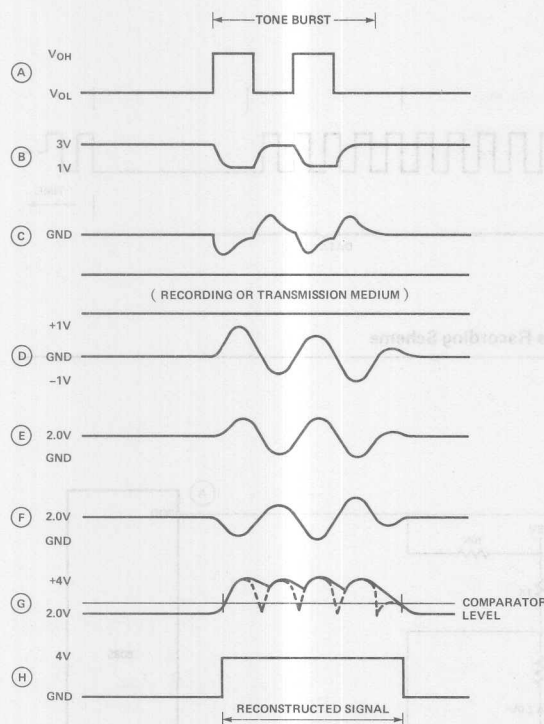


Figure 9. Analog Signal Waveforms

After the CRT software analysis, the tape routines are almost trivial. TAPEO is a subroutine for outputting the contents of register C to a cassette recorder. TAPEIN reads 8 bits into register C.

### OUTPUT ROUTINE

TAPEO calls a subroutine named BURST three times for each bit. If A<sub>6</sub> (the SOD enable bit) is set when BURST is called, a square-wave tone burst will be transmitted. If A<sub>6</sub> is not set, BURST simply delays for exactly the same amount of time before returning. The three calls are used to, respectively, output the initial burst, output the data burst/space, and create the space at the end of each bit. Nine bits will be output: the eight data bits (LSB first) followed by a zero bit. The start of the initial burst of the trailing zero is needed to mark the end of the final space of the preceding data bit.

Start each bit by outputting a tone burst:

```
TAPEO: MVI    B, 9
T01:   MVI    A, 0C0H
      CALL    BURST
```

Rotate register C through CY:

```
MOV    A, C
RAR
MOV    C, A
```

Move CY to the SOD enable bit position, A<sub>6</sub>. Simultaneously set A<sub>7</sub> to one, and clear all other bits. Output a tone burst or space, depending on the previous contents of CY:

```
MVI    A, 01H
RAR
RAR
CALL    BURST
```

Clear the accumulator, and output a space:

```
XRA    A
CALL    BURST
```

Keep cycling until the full 9-bit sequence is finished:

```
DCR    B
JNZ    T01
RET
```

The BURST subroutine executes the SIM instruction CYCNO times, at intervals of 29 + 14 (HALFCYC) machine cycles. In between each SIM, bit A<sub>7</sub> is complemented. CYCNO should be an even number. If A<sub>6</sub> is set upon calling BURST a square-wave will be created. Otherwise, the same code sequence is followed but SOD does not change — thus a space results.

```
BURST: MVI    D, CYCNO    <7>
BU1:   SIM                      <4>
      MVI    E, HALFCYC    <7>
BU2:   DCR    E            <4>
      JNZ    BU2            <7/10>
      XRI    80H            <7>
      DCR    D              <4>
      JNZ    BU1            <7/10>
      RET                    <10>
```

### INPUT ROUTINE

TAPEIN uses a subroutine called BITIN to move the data at the SID pin into the CY. The maximum rate at which SID is read is limited by a delay loop in BITIN.

Initialize the bit counter and the register D, which will keep track of the tone burst time. If a tone

burst is being received when TAPEIN is called, wait until the burst is over:

```

TAPEIN: MVI    B, 8
        MVI    D, 00H
TI1:    CALL   BITIN
        JC     TI1
        CALL   BITIN
        JC     TI1

```

(Throughout this subroutine, a level transition is recognized only after it has been read once initially and then verified on the next reading. This provides some degree of software noise immunity.) Now await the start of the next burst:

```

TI2:    CALL   BITIN
        JNC    TI2
        CALL   BITIN
        JNC    TI2

```

The next burst has now arrived. Keep reading the SID pin, decrementing register D (thus making it more negative), each cycle until the pause is detected:

```

TI3:    DCR    D
        CALL   BITIN
        JC     TI3
        CALL   BITIN
        JC     TI3

```

Now continue reading the SID pin, incrementing the D register (back towards zero), each cycle until the next burst is received:

```

TI4:    INR    D
        CALL   BITIN
        JNC    TI4
        CALL   BITIN
        JNC    TI4

```

Now, if the burst lasted longer than the space, D was not incremented all the way back to zero; it is still negative. If the space was longer, D was incremented up through zero; it is now positive. In other words, the sign bit of D will now correspond to the data bit that would lead to each of these results. Move the sign bit into the CY, then rotate it into register C:

```

MOV     A, D
RAL
MOV     A, C
RAR
MOV     C, A
MVI    D, 00H

```

Continue until the last bit has been received:

```

DCR     B
JNZ     TI3
RET

```

(Notice that the first half of this subroutine is incorporated in the second half. In fact, the assembled listing included in the Appendix makes use of this fact to eliminate 24 bytes of duplicated code.)

BITIN waits a short time in order to regulate the sampling rate, then reads SID and moves the data bit into the CY:

```

BITIN: MVI     E, CKRATE
BI1:    DCR     E
        JNZ     BI1
        RIM
        RAL
        RET

```

The tone burst frequency and duration, and the TAPEIN sampling rate are determined by HALFCYC, CYCNO, and CKRATE. Tables 2 and 3 give typical values.

Table 2  
EXAMPLE COMBINATIONS OF HALFCYC AND CYCNO.  
ALL VALUES IN DECIMAL

APPROXIMATE TONE FREQUENCY	CORRESPONDING HALFCYC VALUE	RESULTING DATA RATE			
		8 4	20 10	100 50	CYCNO CYC/BURST
500 Hz	217	42	17	3.3	bps
1 kHz	108	83	33	6.6	bps
2 kHz	53	166	66	13	bps
5 kHz	20	414	166	33	bps
10 kHz	9	826	330	66	bps

Table 3  
MAXIMUM SAMPLING RATES  
FOR VARIOUS VALUES OF  
CKRATE

CKRATE VALUE	SAMPLING RATE (INCLUDING CALL & RET)
1	17.6 $\mu$ sec
20	104 $\mu$ sec
80	378 $\mu$ sec
250	1.14 msec

HL must be set to the start of the desired block, and the recorder turned on manually. Successive bytes will be recorded until the end of that page, i.e., until L is incremented to zero. The playback routine requires presetting HL to the target address and turning on the recorder before PLAYBK is called. These routines incorporate a long tone burst before each data block to allow a recorder with Automatic Gain Control to stabilize before the data starts.

### ADDITIONAL COMMENTS

The two design examples given so far were built up using an SDK-85 System Design Kit. Both hardware interfaces were wire-wrapped on the ample breadboarding area provided on the board. The connections between SID and SOD and the on-board TTY interface were broken, so as not to affect the 8085 I/O electrical characteristics.

The CRT interface was tested with a Beehive Mini-Bee II Terminal in the full duplex mode at each of its 14 possible transmission rates, from 110 to 9600 baud. It was also checked out at 19200 baud using a Beehive B-100 terminal. In addition, the software was exercised using an SBC 80/20 system as a variable baud rate character generator and receiver.

An additional advantage to having software selectable communications rates is that it would be possible to communicate with several system peripherals, each at its own preferred rate, without having to duplicate hardware. For example, the addition of a single 7408 AND gate and an output port (such as on the 8155) would make it possible to use the same two RS-232 circuits to interface with up to seven I/O devices (see Figure 10). Three of the MC1488 drivers have Enable inputs which can be controlled by the output port. One AND gate can be used to buffer the SOD line and drive the MC1488 Data inputs. The rest of the 7408 can be configured as a four input AND gate. This would act as an inverted logic OR gate to reduce the four MC1489 receiver outputs to a single line, which could be read by the SID. This assumes that only one input device (CRT, PTR) at a time will be used (which is usually the case in a non-time shared, interactive application), and that the unused devices are transmitting a logic one level (which should also be the case).

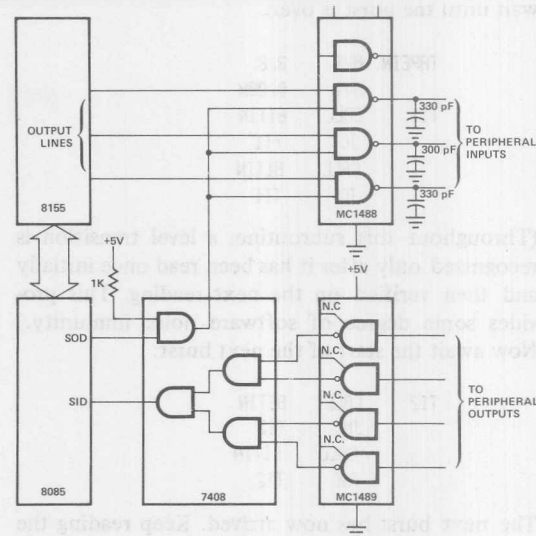


Figure 10. Interfacing 8085 to Multiple Peripherals

The software needed to support additional peripherals would be simple and straightforward. A routine intended to dump a section of memory to a paper tape punch, for example, would first have to store BITTIME and HALFBIT somewhere (perhaps on stack), load the variables with new parameters corresponding to the paper tape punch rate, and then write a bit pattern to the output port which would disable the console driver and enable the punch (and perhaps a typewriter). After the dump was over, the original time parameters and driver status would be restored.

As explained before, the BRID routine computed rate parameters based on the fact that an ASCII "space" character resulted in a zero level 6 bits long. Conceivably, some obscure peripherals might produce a transient between successive zero bits. (This might be the case, for example, if the signal was produced by mechanical rather than electronic means.) If so, the BRID algorithm used here probably would not work reliably. Once the two time parameters were identified, though, COUT and CIN could still be used. An alternate algorithm for baud rate identification would require a table in ROM (note the fifth and final R/S-I/O-M/D permutation). This table would contain a list of delay parameters corresponding to the standard transmis-

Depending on the recorder quality, the tone burst frequency and duration can be optimized for higher data rates by modifying HALFCYC and CYCNO. If so, CKRATE should also be reduced, so that between about 10 and 80 data samplings are made during a single (one-third width) tone burst. At greatly increased frequencies, some of the

Funny things happen when recording and playing back a page of RAM which includes the subroutine stack. Eventually, `PLAYBK` will start writing over the data at the top of the stack, destroying the subroutine traceback sequence. The next `RET` instruction will then cause a jump to a place where you'd rather not be.

The printout reproduced in the Appendix includes the assembled listings for the CRT and magnetic tape interfaces discussed in this application note. The object code produced was programmed into an 8755 EPROM, which was installed in the expansion PROM socket of the SDK-85 board. Some very minor differences exist between this listing and the code segments presented earlier, which were written for maximum clarity.

## Appendix 1.1 — Temperature Sensor Code

ASM80 : F1:TEST SRC MOD85

ISIS-II 8888/8885 MACRO ASSEMBLER, V2.0

MODULE PAGE 1

LOC	OBJ	SEQ	SOURCE STATEMENT
		1 ;	
		2 ;	
026C		3	HXDSP EQU 026CH ;EXPAND HEX TO DISPLAY, SDK MONITOR ROUTINE
02B7		4	OUTPUT EQU 02B7H ;OUTPUT TO DISPLAY, SDK MONITOR ROUTINE
05F1		5	DELAY EQU 05F1H ;DELAY DISPLAY, SDK MONITOR ROUTINE
		6 ;	
2000		7	ORG 2000H
		8 ;	
		9 ;	
		10 ;	
		11 ;	
2000 31C820		12	LXI SP,20C8H ; INITIALIZE STACKPOINTER
2003 F3		13	DI ; DISABLE INTERRUPTS
		14 ;	
		15 ;	INITIALIZE COUNTER IN 8155 FOR COUNTDOWN MODE. LOAD COUNTER
		16 ;	WITH HIGHEST VALUE (3FFF).
		17 ;	
2004 3EBF		18	MVI A,0BFH ;
2006 D325		19	OUT 25H ; ADDRESS FOR TOP HALF OF COUNTER
2008 3EFF		20	MVI A,0FFH ;
200A D324		21	OUT 24H ; " " LOWER HALF OF COUNTER
200C 3EC0		22	MVI A,0C0H ;
200E D320		23	OUT 20H ; COUNT DOWN MODE START
		24 ;	
		25 ;	PULSE THE ONE SHOT WITH A POSITIVE GOING PULSE ON THE S00
		26 ;	OUTPUT PIN OF THE 8885.
		27 ;	
2010 3EC8		28	MVI A,0C8H ;
2012 30		29	SIM ; OUTPUT A HIGH ON S00 LINE
2013 3E48		30	MVI A,48H ;
2015 30		31	SIM ; OUTPUT A LOW ON S00 LINE
2016 FB		32	EI ; ENABLE INTERRUPTS(AFTER PULSE)
		33 ;	
		34 ;	IDLE UNTIL ONESHOT INTERRUPTS THE RST 6.5 PIN ON THE 8885.
		35 ;	
2017 00		36	NPO: NOP ;
2018 C31720		37	JMP NPO ; IDLE UNTIL INTERRUPT
		38 ;	
		39 ;	AFTER INTERRUPT, STOP COUNTER AND READ IN FINAL COUNT FROM
		40 ;	8155. STORE IN REGISTER PAIR BC.
		41 ;	
201B 3E40		42	OUT: MVI A,40H ;
201D D320		43	OUT 20H ; STOP COUNTER
201F DB24		44	IN 24H ;
2021 4F		45	MOV C,A ; STORE LOWER ORDER BYTE IN C
2022 DB25		46	IN 25H ;
2024 47		47	MOV B,A ; STORE HIGHER ORDER BYTE IN B
2025 263F		48	MVI H,3FH ; LOAD HL WITH FULL START COUNT
2027 2EFF		49	MVI L,0FFH ;
		50 ;	
		51 ;	ADJUST THE COUNT VALUE IN REGISTER BC TO REPRESENT ACTUAL
		52 ;	COUNT. (SEE TEXT FOR EXPLANATION).

LOC	OBJ	SEQ	SOURCE STATEMENT
		53 ;	
2829	CD6820	54	CALL ADJUST ; CONVERTS 8155 COUNT TO ACTUAL COUNT
		55 ;	
		56 ;	SETUP INITIALIZATION FOR SEARCH ROUTINE. ROUTINE LOOKS FOR TEMPERATURE
		57 ;	RANGE OF COUNT (SEE TEXT). SEARCH ONLY FOR UPPER HALF TO SIMPLIFY CODE.
		58 ;	
282C	2E80	59	MVI L, 80H ; SET HL TO BEGINNING OF SEARCH
282E	2620	60	MVI H, 20H ; STRING IN MEMORY.
2830	B0	61	ORA B ; CLEAR CARRY FOR ROUTINE
2831	78	62	MOV A, B ; PLACE B INTO ACCUMULATOR
2832	0E01	63	MVI C, 1H ; SET TIMES THROUGH SEARCH
2834	CD9220	64	CALL SEARCH ; LOOKS FOR TEMP RANGE COUNT IS IN
		65 ;	
		66 ;	CHECK IF SEARCH WAS SUCCESSFUL. IF NOT THEN OUTSIDE ACCEPTABLE
		67 ;	RANGE.
		68 ;	
2837	3E80	69	MVI A, 80H ; DID L FIND LESS THAN AT
2839	AD	70	XRA L ; AT BEGINNING OF STRING?
283A	C9AF20	71	JZ TLOW ; TEMP BELOW ALLOWED LIMITS, SET PORT A
283D	3E80	72	MVI A, 80H ; DID C GET DECREMENTED?
283F	B9	73	CMP C ; IF SO, SEARCH DID NOT FIND
2840	CAB820	74	JZ THIGH ; TEMP ABOVE LIMITS, SET PORT B
		75 ;	
		76 ;	SOFTWARE MAP THE MATCH TO A TEMPERATURE IN DEGREES C BY ADDING
		77 ;	10 TO SEARCH ADDRESS. PLACE TEMPERATURE IN REGISTER E.
		78 ;	
2843	3E8A	79	MVI A, 0AH ; SHIFT HL BY 10 (SOFTWARE MAP)
2845	85	80	ADD L
2846	6F	81	MOV L, A
2847	5E	82	MOV E, M ; READ IN TEMPERATURE
		83 ;	
		84 ;	SET UP INITIALIZATION FOR DISPLAYING TEMPERATURE USING SDK
		85 ;	MONITOR ROUTINES. FIRST EXPAND DE REGISTER AND THEN DISPLAY
		86 ;	FOR DELAY PERIOD.
		87 ;	
2848	0600	88	MVI B, 00H ; CLEAR DOT AT ADDRESS FIELD
284A	CD6C02	89	CALL HXDSP ; CALL EXPAND
284D	3E00	90	MVI A, 00H
284F	CD8702	91	CALL OUTPUT ; OUTPUT TO SDK DISPLAY
2852	11FF00	92	LXI D, 0FFH ; SET DELAY PERIOD
2855	CD105	93	CALL DELAY ; DISPLAY FOR DELAY PERIOD
2858	CF	94	RST 1 ; SOFTWARE RESTART
		95 ;	
		96 ;	SUBROUTINES
		97 ;	
28AF		98 ORG	20AFH
		99 ;	
		100 ;	
28AF	3E03	101 TLOW:	MVI A, 03H
28B1	D320	102	OUT 20H
28B3	3EFF	103	MVI A, 0FFH ; SET PORT A AS 1'S
28B5	D321	104	OUT 21H
28B7	CF	105	RST 1
		106 ;	
		107 ;	

LOC	OBJ	SEQ	SOURCE STATEMENT	THRESHOLD	330000	932	100	001
2088	3E03	108	THIGH: MVI A,03H					
208A	D320	109	OUT 20H					
208C	3EFF	110	MVI A,0FFH ;SET PORT B AS 1'S					
208E	D322	111	OUT 22H					
20C0	CF	112	RST 1					
		113	;					
		114	;					
2092		115	ORG 2092H					
		116	;					
		117	;					
2092	BE	118	SEARCH: CMP M					
2093	D8	119	RC					
2094	Z3	120	INX H ;ELSE INCREMENT POINTER					
2095	BE	121	CMP M ;COMPARE 2ND BYTE					
2096	D8	122	RC					
2097	Z3	123	INX H					
2098	BE	124	CMP M ;COMPARE 3RD BYTE					
2099	D8	125	RC					
209A	Z3	126	INX H					
209B	BE	127	CMP M ;COMPARE 4TH BYTE					
209C	D8	128	RC					
209D	Z3	129	INX H					
209E	BE	130	CMP M ;COMPARE 5TH BYTE					
209F	D8	131	RC					
20A0	Z3	132	INX H					
20A1	BE	133	CMP M ;COMPARE 6TH BYTE					
20A2	D8	134	RC					
20A3	Z3	135	INX H					
20A4	BE	136	CMP M ;COMPARE 7TH BYTE					
20A5	D8	137	RC					
20A6	Z3	138	INX H					
20A7	BE	139	CMP M ;COMPARE 8TH BYTE					
20A8	D8	140	RC					
20A9	Z3	141	INX H					
20AA	00	142	DCR C ;HAS ENTIRE BLOCK BEEN					
20AB	C29220	143	JNZ SEARCH ;SEARCHED? IF 50 SET NO					
20AE	C9	144	RET ;LESS THAN AND RETURN.					
		145	;					
		146	RESTART 6.5 JUMP ADDRESS					
		147	;					
20CE		148	ORG 20CEH					
		149	;					
		150	;					
20CE	C31B20	151	JMP CNTU					
		152	;					
		153	;					
		154	;					
		155	;					
		156	;					
		157	;					
		158	SEARCH COMPARE DATA STRING (SEE TEXT)					
		159	;					
		160	;					
2080		161	ORG 2080H					
		162	;					

LOC	OBJ	SEQ	SOURCE STATEMENT
2080	35	163 ;	
2081	36	164	DB 35H, 36H, 37H, 38H, 39H, 3AH, 3BH, 3CH
2082	37		
2083	38		
2084	39		
2085	3A		
2086	3B		
2087	3C		
		165 ;	
		166 ;	SOFTWARE MAP TO TEMPERATURE
		167 ;	
2088		168 ORG	2088H
		169 ;	
		170 ;	
2088	21	171	DB 21H, 23H, 25H, 28H, 31H, 35H, 39H
208C	23		
208D	25		
208E	28		
208F	31		
2090	35		
2091	39		
		172 ;	
		173 ;	
2060		174 ORG	2060H
		175 ;	
		176 ;	
		177 ;	SUBROUTINE ADJUST FOR COUNT IN 8155
		178 ;	
2060	78	179	ADJUST: MOV A, B ; LOAD ACCUMULATOR WITH UPPER HALF
2061	E63F	180	ANI 3FH ; RESET UPPER TWO BITS; CLEAR CARRY
2063	1F	181	RAR ; ROTATE RIGHT THROUGH CARRY
2064	47	182	MOV B, A ; STORE SHIFTED VALUE BACK IN B
2065	79	183	MOV A, C ; LOAD ACCUMULATOR WITH LOWER HALF
2066	1F	184	RAR ; ROTATE WITH CARRY RIGHT
2067	4F	185	MOV C, A ; STORE SHIFTED VALUE IN C
2068	D0	186	RNC ; 1ST HALF OR SECOND? IF SECOND RETURN
2069	3F	187	CNC ; CLEAR CARRY
206A	7C	188	MOV A, H ; OBTAIN ONE HALF OF FULL COUNT.
206B	1F	189	RAR ; IF HL IS ODD THIS CONTAINS
206C	67	190	MOV H, A ; ONE HALF(FULL COUNT-1), WHICH
206D	7D	191	MOV A, L ; IS CORRECT.
206E	1F	192	RAR
206F	6F	193	MOV L, A
2070	09	194	DAD B ; DOUBLE PRECISION ADD
2071	44	195	MOV B, H ; RESTORE BC REGISTERS WITH COUNT
2072	4D	196	MOV C, L
2073	C9	197	RET
		198 ;	
		199 ;	
		200	END

PUBLIC SYMBOLS

EXTERNAL SYMBOLS

USER SYMBOLS

ADJUST A 2060 CNTU A 201B DELAY A 05F1 HXDSP A 026C NPO A 2017 OUTPUT A 0267 SEARCH A 2092  
 THIGH A 2068 TLOW A 20AF

ASSEMBLY COMPLETE, NO ERRORS



LOC	OBJ	SEQ	SOURCE STATEMENT
		1	
		2 ;	THE FOLLOWING PROGRAMS AND SUBROUTINES ARE DESCRIBED IN DETAIL
		3 ;	IN INTEL CORPORATION'S APPLICATION NOTE AP-29, "USING THE 8085
		4 ;	SERIAL I/O LINES". THE FIRST SECTION IS A GENERAL PURPOSE CRT
		5 ;	INTERFACE WITH AUTOMATIC BAUD RATE IDENTIFICATION; THE SECOND
		6 ;	SECTION IS A MAGNETIC TAPE INTERFACE FOR STORING DATA ON CASSETTE
		7 ;	TAPE. THE CODE PRESENTED HERE IS ORIGINATED AT LOCATION 000H,
		8 ;	AND MIGHT BE PART OF AN EXPANSION FROM IN AN INTEL SDK-85
		9 ;	SYSTEM DESIGN KIT.
		10 :	
		11	
		12	
20C8		13	BITTIME EQU 20C8H ; ADDRESS OF STORAGE FOR COMPUTED BIT DELAY
20CA		14	HALFBIT EQU 20CAH ; ADDRESS OF STORAGE FOR HALF BIT DELAY
0008		15	BITSO EQU 11 ; DATA BITS PUT OUT (INCLUDING TWO STOP BITS)
0009		16	BITSI EQU 9 ; DATA BITS TO BE RECIEVED (INCLUDING ONE STOP BIT)
		17	
0800		18	ORG 800H ; STARTING ADDRESS OF SDK-85 EXPANSION PROM
		19	
		20 ;	CRTTST CRT INTERFACE TEST. WHEN CALLED, AWAITS THE SPACE BAR BEING PRESSED ON
		21 ;	THE SYSTEM CONSOLE, AND THEN RESPONDS WITH A DATA RATE VERIFICATION
		22 ;	MESSAGE. THERE AFTER, CHARACTERS TYPED ON THE KEYBOARD ARE ECHOED
		23 ;	ON THE DISPLAY TUBE. WHEN A BREAK KEY IS TYPED, THE ROUTINE IS
		24 ;	RE-STARTED, ALLOWING A DIFFERENT BAUD RATE TO BE SELECTED ON THE CRT.
0800 31C020		25	CRTTST LXI SP, 20C0H
0803 3EC0		26	CRT1: MVI A, 0C0H ; SOD MUST BE HIGH BETWEEN CHARACTERS
0805 30		27	SIM
0806 CD1A08		28	CALL BRID ; IDENTIFY DATA RATE USED BY TERMINAL
0809 CD4708		29	CALL SIGNON ; OUTPUT SIGNON MESSAGE AT RATE DETECTED
080C CD8A08		30	ECHO: CALL CIN ; READ NEXT KEYSTROKE INTO REGISTER C
080F 79		31	MOV A, C
0810 B7		32	ORA A ; CHECK IF CHARACTER WAS A <BREAK> (ASCII 00H)
0811 CA0308		33	JZ CRT1 ; IF SO, RE-IDENTIFY DATA RATE
		34	; THIS ALLOWS ANOTHER RATE TO BE SELECTED ON CRT
0814 CD6908		35	CALL COUT ; OTHERWISE COPY REGISTER C TO THE SCREEN
0817 C30C08		36	JMP ECHO ; CONTINUE INDEFINITELY (UNTIL BREAK)
		37	
		38 ;	BRID BAUD RATE IDENTIFICATION SUBROUTINE
		39 ;	EXPECTS A <CR> (ASCII 20H) TO BE RECIEVED FROM THE CONSOLE.
		40 ;	THE LENGTH OF THE INITIAL ZERO LEVEL (SIX BITS WIDE) IS MEASURED
		41 ;	IN ORDER TO DETERMINE THE DATA RATE FOR FUTURE COMMUNICATIONS.
081A 20		42	BRID: RIM ; VERIFY THAT THE "ONE" LEVEL HAS BEEN ESTABLISHED
081B B7		43	ORA A ; AS THE CRT IS POWERING UP
081C F21A08		44	JP BRID
081F 20		45	BRI1: RIM ; MONITOR SID LINE STATUS
0820 B7		46	ORA A
0821 FA1F08		47	JM BRI1 ; LOOP UNTIL START BIT IS RECIEVED
0824 21FAFF		48	LXI H, -6 ; BIAS COUNTER USED IN DETERMINING ZERO DURATION
0827 1E04		49	BRI3: MVI E, 04H
0829 10		50	BRI4: DCR E ; S3 MACHINE CYCLE DELAY LOOP
082A C22908		51	JNZ BRI4
082D 23		52	INX H ; INCREMENT COUNTER EVERY 84 CYCLES WHILE SID IS LOW
082E 20		53	RIM

002F B7	54	ORA	A	
0030 F22708	55	JP	BR13	
	56			; <HL> NOW CORRESPONDS TO INCOMING DATA RATE
0033 E5	57	PUSH	H	; SAVE COUNT FOR HALFBIT TIME COMPUTATION
0034 24	58	INR	H	; BITTIME IS DETERMINED BY INCREMENTING
0035 2C	59	INR	L	; \ H AND L INDIVIDUALLY
0036 22C820	60	SHLD	BITTIME	
0039 E1	61	POP	H	; RESTORE COUNT FOR HALFBIT DETERMINATION
003A B7	62	ORA	A	; CLEAR CARRY
003B 7C	63	MOV	A, H	; ROTATE RIGHT EXTENDED <HL>
003C 1F	64	RAR		; \ TO DIVIDE COUNT BY 2
003D 67	65	MOV	H, A	
003E 7D	66	MOV	A, L	
003F 1F	67	RAR		
0040 6F	68	MOV	L, A	
0041 24	69	INR	H	; PUT H AND L IN PROPER FORMAT FOR DELAY
0042 2C	70	INR	L	; \ SEGMENTS (INCREMENT EACH)
0043 22CA20	71	SHLD	HALFBIT	; SAVE AS HALF-BIT TIME DELAY PARAMETER
0046 C9	72	PET		
	73			
	74	; SIGNON WRITES A SIGN-ON MESSAGE TO THE CRT AT WHAT SHOULD BE THE CORRECT RATE.		
	75	; IF THE MESSAGE IS UNINTELLIGIBLE. WELL, SO IT GOES.		
0047 215508	76	SIGNON:	LXI	A, STRING ; LOAD START OF SIGN-ON MESSAGE
004A 4E	77	S1:	MOV	C, M ; GET NEXT CHARACTER
004B AF	78	XRA	A	; CLEAR ACCUMULATOR
004C B1	79	ORA	C	; CHECK IF CHARACTER IS END OF STRING
004D C8	80	RZ		; RETURN IF SIGN-ON COMPLETE
004E CD6908	81	CALL	COUT	; ELSE OUTPUT CHARACTER TO CRT
0051 23	82	INX	H	; INDEX POINTER
0052 C34A08	83	JMP	S1	; ECHO NEXT CHARACTER
	84			
0055 8D	85	STRING:	DB	0DH, 0AH ; <CR><LF>
0056 0A				
0057 42415544	86		DB	'BAUD RATE CHECK'
005B 20524154				
005F 45204348				
0063 45434B				
0065 0D	87		DB	0DH, 0AH ; <CR><LF>
0067 0A				
0068 00	88		DB	00H ; END-OF-STRING ESCAPE CODE
	89			
	90	COUT	CONSOLE OUTPUT SUBROUTINE	
	91		; WRITES THE CONTENTS OF THE C REGISTER TO THE CRT DISPLAY SCREEN	
0069 F3	92	COUT:	DI	
006A C5	93		PUSH	B
006B E5	94		PUSH	H
006C 0608	95	MVI	B, BIT50	; SET NUMBER OF BITS TO BE TRANSMITTED
006E AF	96	XRA	A	; CLEAR CARRY
006F 3E80	97	C01:	MVI	A, 80H ; SET WHAT WILL BECOME SOD ENABLE BIT
0071 1F	98		RAR	; MOVE CARRY INTO SOD DATA BIT OF ACC
0072 30	99		SIM	; OUTPUT DATA BIT TO SOD
0073 2AC820	100	LHLD	BITTIME	
0076 2D	101	C02:	DCR	L ; WAIT UNTIL APPROPRIATE TIME HAS PASSED

```

0077 C27608      102      JNZ      C02
007A 25          103      DCR      H
007B C27608      104      JNZ      C02
007E 37          105      STC              ;SET WHAT WILL EVENTUALLY BECOME A STOP BIT
007F 79          106      MOV      A,C      ;ROTATE CHARACTER RIGHT ONE BIT,
0080 1F          107      RAR              ;\ MOVING NEXT DATA BIT INTO CARRY
0081 4F          108      MOV      C,A
0082 05          109      DCR      B      ;CHECK IF CHARACTER (AND STOP BIT(S)) DONE
0083 C26F08      110      JNZ      C01      ;IF NOT, OUTPUT CURRENT CARRY
0086 E1          111      POP      H      ;RESTORE STATUS AND RETURN
0087 C1          112      POP      B
0088 FB          113      EI
0089 C9          114      RET
                115
                116 ;CIN  CONSOL INPUT SUBROUTINE WAITS FOR A KEYSTROKE AND
                117 ;      RETURNS WITH 8 BITS IN REG C.
008A F3          118 CIN:  DI
008B E5          119      PUSH     H
008C 0609        120      MVI      B,BITSI ;DATA BITS TO BE READ (LAST RETURNED IN CY)
008E 20          121 C11:  RIM              ;WAIT FOR SYNC BIT TRANSITION
008F B7          122      ORA      A
0090 FA8E08      123      JM       C11
0093 2ACA20      124      LHLD     HALFBIT
0096 20          125 C12:  DCR      L      ;WAIT UNTIL MIDDLE OF START BIT
0097 C29608      126      JNZ      C12
009A 25          127      DCR      H
009B C29608      128      JNZ      C12
009E 2AC820      129 C13:  LHLD     BITTIME ;WAIT OUT BIT TIME
00A1 20          130 C14:  DCR      L
00A2 C2A108      131      JNZ      C14
00A5 25          132      DCR      H
00A6 C2A108      133      JNZ      C14
00A9 20          134      RIM              ;CHECK SID LINE LEVEL
00AA 17          135      RAL              ;DATA BIT IN CY
00AB 05          136      DCR      B      ;DETERMINE IF THIS IS FIRST STOP BIT
00AC CAB608      137      JZ       C15      ;IF SO, JUMP OUT OF LOOP
00AF 79          138      MOV      A,C      ;ELSE ROTATE INTO PARTIAL CHARACTER IN C
00B0 1F          139      RAR              ;ACC HOLDS UPDATED CHARACTER
00B1 4F          140      MOV      C,A
00B2 00          141      NOP              ;EQUALIZES COUT AND CIN LOOP TIMES
00B3 C29E08      142      JMP      C13
00B6 E1          143 C15:  POP      H
00B7 FB          144      EI
00B8 C9          145      RET              ;CHARACTER COMPLETE
                146
00B9 C9          147 ;*****
00BA C9          148 ;
00BB C9          149 ; THE FOLLOWING CODE IS USED BY THE CASSETTE INTERFACE.
00BC C9          150 ; SUBROUTINES TAPEO AND TAPEIN ARE USED RESPECTIVELY
00BD C9          151 ; TO OUTPUT OR RECEIVE AN EIGHT BIT BYTE OF DATA. REGISTER C
00BE C9          152 ; HOLDS THE DATA IN EITHER CASE. REGISTERS A,B,&C ARE ALL DESTROYED.
00BF C9          153 CYCNO EQU 16 ;TWICE THE NUMBER OF CYCLES PER TONE BURST
00C0 C9          154 HALFCYC EQU 30 ;DETERMINES TONE FREQUENCY

```

LOC	OBJ	SEQ	SOURCE STATEMENT
0016		155	OKRATE EQU 22 ;SETS SAMPLE RATE
00FA		156	LEADER EQU 250 ;NUMBER OF SUCCESSIVE TONE BURSTS COMPRISING LEADER
00FA		157	LDROCK EQU 250 ;USED IN PLAYEK TO VERIFY PRESENCE OF LEADER
		158	
		159	BLKROD OUTPUTS A VERY LONG TONE BURST (<LEADER> TIMES
		160	THE NORMAL BURST DURATION) TO ALLOW RECORDER ELECTRONICS
		161	AND AGC TO STABILIZE, THEN OUTPUTS THE REMAINDER OF THE
		162	256 BYTE PAGE POINTED TO BY <H>, STARTING AT BYTE <L>.
00B9 00FA		163	BLKROD: MVI C,LEADER;SET UP LEADER BURST LENGTH
00BB 3EC0		164	MVI A,00AH ;SET ACCUMULATOR TO RESULT IN TONE BURST
00BD CDF008		165	BR1: CALL BURST ;OUTPUT TONE
00C0 00		166	DCR C
00C1 C2BD08		167	JNZ BR1 ;SUSTAIN LEADER TONE
00C4 AF		168	XRA A ;CLEAR ACCUMULATOR & OUTPUT SPACE, SO THAT
00C5 CDF008		169	CALL BURST ;\ START OF FIRST DATA BYTE CAN BE DETECTED
00C8 4E		170	BR2: MOV C,M ;GET DATA BYTE TO BE RECORDED
00C9 CDD108		171	CALL TAPEO ;OUTPUT REGISTER C TO RECORDER
00CC 2C		172	INR L ;POINT TO NEXT BYTE
00CD C2C008		173	JNZ BR2
00D0 C9		174	RET ;AFTER BLOCK IS COMPLETE
		175	
		176	
		177	TAPEO OUTPUTS THE BYTE IN REGISTER C TO THE RECORDER.
		178	REGISTERS A,B,C,D,&E ARE ALL USED.
00D1 F3		179	TAPEO: DI
00D2 05		180	PUSH D ;D&E USED AS COUNTERS BY SUBROUTINE BURST
00D3 0609		181	MVI B,9 ;WILL RESULT IN 8 DATA BITS AND ONE STOP BIT
00D5 AF		182	T01: XRA A ;CLEAR ACCUMULATOR
00D6 3EC0		183	MVI A,00AH ;SET ACCUMULATOR TO CAUSE A TONE BURST
00D8 CDF008		184	CALL BURST
00DB 79		185	MOV A,C ;MOVE NEXT DATA BIT INTO THE CARRY
00DC 1F		186	RAR
00DD 4F		187	MOV C,A ;CARRY WILL BECOME SOD ENABLE IN BURST ROUTINE
00DE 3E01		188	MVI A,01H ;SET BIT TO BE REPEATEDLY COMPLEMENTED IN BURST
00E0 1F		189	RAR
00E1 1F		190	RAR
00E2 CDF008		191	CALL BURST ;OUTPUT EITHER A TONE OR A PAUSE
00E5 AF		192	XRA A ;CLEAR ACCUMULATOR
00E6 CDF008		193	CALL BURST ;OUTPUT PAUSE
00E9 05		194	DCR B
00EA C2D508		195	JNZ T01 ;REPEAT UNTIL BYTE FINISHED
00ED D1		196	POP D ;RESTORE STATUS AND RETURN
00EE FB		197	EI
00EF C9		198	RET
		199	
00F0 1610		200	BURST: MVI D,CYCNO ;SET NUMBER OF CYCLES
00F2 30		201	BU1: SIM ;COMPLEMENT SOD LINE IF SOD ENABLE BIT SET
00F3 1E1E		202	MVI E,HALFCYC
00F5 1D		203	BU2: DCR E ;REGULATE TONE FREQUENCY
00F6 C2F508		204	JNZ BU2
00F9 EE00		205	XRI 80H ;COMPLEMENT SOD DATA BIT IN ACCUMULATOR
00FB 15		206	DCR D
00FC C2F208		207	JNZ BU1 ;CONTINUE UNTIL BURST (OR EQUIVILENT PAUSE) FINISHED

LOC	OBJ	SEQ	SOURCE STATEMENT
08FF	C9	208	RET
		209	
		210	;PLAYBK WAITS FOR THE LONG LEADER BURST TO ARRIVE, THEN CONTINUES
		211	; READING BYTES FROM THE RECORDER AND STORING THEM
		212	; IN MEMORY STARTING AT LOCATION <HL>.
		213	; CONTINUES UNTIL THE END OF THE CURRENT PAGE (<L>=0FFH) IS REACHED.
0900	0EFA	214	PLAYBK: MVI C,LDRCBK ;<LDRCBK> SUCCESSIVE HIGHS MUST BE READ
0902	CD3D09	215	PB1: CALL BITIN ;\ TO VERIFY THAT THE LEADER IS PRESENT
0905	D20009	216	JNC PLAYBK ; \ AND ELECTRONICS HAS STABILIZED
0908	0D	217	DCR C
0909	C20209	218	JNZ PB1
090C	CD1509	219	PB2: CALL TAPEIN ;GET DATA BYTE FROM RECORDER
090F	71	220	MOV M,C ;STORE IN MEMORY
0910	2C	221	INR L ;INCREMENT POINTER
0911	C20C09	222	JNZ PB2 ;REPEAT FOR REST OF CURRENT PAGE
0914	C9	223	RET
		224	
		225	;TAPEIN CASSETTE TAPE INPUT SUBROUTINE. READS ONE BYTE OF DATA
		226	; FROM THE RECORDER INTERFACE AND RETURNS WITH THE BYTE IN REGISTER C.
0915	0609	227	TAPEIN: MVI B,9 ;READ EIGHT DATA BITS
0917	1600	228	TI1: MVI D,00H ;CLEAR UP/DOWN COUNTER
0919	15	229	TI2: DCR D ;DECREMENT COUNTER EACH TIME ONE LEVEL IS READ
091A	CD3D09	230	CALL BITIN
091D	DA1909	231	JC TI2 ;REPEAT IF STILL AT ONE LEVEL
0920	CD3D09	232	CALL BITIN
0923	DA1909	233	JC TI2
0926	14	234	TI3: INR D ;INCREMENT COUNTER EACH TIME ZERO IS READ
0927	CD3D09	235	CALL BITIN
092A	D22609	236	JNC TI3 ;REPEAT EACH TIME ZERO IS READ
092D	CD3D09	237	CALL BITIN
0930	D22609	238	JNC TI3
0933	7A	239	MOV A,D
0934	17	240	RAL ;MOVE COUNTER MOST SIGNIFICANT BIT INTO CARRY
0935	79	241	MOV A,C
0936	1F	242	RAR ;MOVE DATA BIT RECEIVED (CY) INTO BYTE REGISTER
0937	4F	243	MOV C,A
0938	05	244	DCR B
0939	C21709	245	JNZ TI1 ;REPEAT UNTIL FULL BYTE ASSEMBLED
093C	C9	246	RET
		247	
093D	1E16	248	BITIN: MVI E,CKRATE
093F	1D	249	BI1: DCR E
0940	C23F09	250	JNZ BI1 ;LIMIT INPUT SAMPLING RATE
0943	20	251	RIM ;SAMPLE SID LINE
0944	17	252	RAL ;MOVE DATA INTO CY BIT
0945	C9	253	RET
		254	
		255	END

PUBLIC SYMBOLS

EXTERNAL SYMBOLS

USER SYMBOLS

BI1	A 093F	BITIN	A 093D	BITSI	A 0909	BITSO	A 090B	BITTIN	A 20C8	BLKRCD	A 0889	BR1	A 088D
BR2	A 09C8	BR11	A 091F	BR13	A 0927	BR14	A 0929	BRID	A 091A	BU1	A 09F2	BU2	A 08F5
BURST	A 09F0	CI1	A 088E	CI2	A 0896	CI3	A 089E	CI4	A 08A1	CI5	A 08B6	CIN	A 088A
CKRATE	A 0916	CO1	A 086F	CO2	A 0876	COUT	A 0869	CRT1	A 0903	CRTTST	A 0900	CYCNO	A 0910
ECHO	A 090C	HALFBI	A 20CA	HALFCY	A 091E	LDARCH	A 09FA	LEADER	A 09FA	PE1	A 0902	PE2	A 090C
PLAYBK	A 0900	SI	A 094A	SIGNON	A 0947	STRNG	A 0855	TAPEIN	A 0915	TAPEO	A 09D1	TI1	A 0917
TI2	A 0919	TI3	A 0926	TO1	A 08D5								

ASSEMBLY COMPLETE, NO ERROR(S)

BI1	249#	250				
BITIN	215	230	232	235	237	248#
BITS1	16#	120				
BITS0	15#	95				
BITTIM	13#	60	100	129		
BLKPCD	163#					
BR1	165#	167				
BR2	170#	173				
BRI1	45#	47				
BRI3	49#	55				
BPI4	50#	51				
BRID	28	42#	44			
BU1	201#	207				
BU2	203#	204				
BURST	165	169	184	191	193	200#
CI1	121#	123				
CI2	125#	126	128			
CI3	129#	142				
CI4	130#	131	133			
CI5	137	143#				
CIN	30	118#				
CKRATE	155#	248				
CO1	97#	110				
CO2	101#	102	104			
COUT	35	81	92#			
CRT1	26#	33				
CRTTST	25#					
CYCNO	153#	200				
ECHO	30#	36				
HALFBI	14#	71	124			
HALFCV	154#	202				
LDRCHK	157#	214				
LEADER	156#	163				
PB1	215#	218				
PB2	219#	222				
PLAYBK	214#	216				
S1	77#	83				
SIGNON	29	76#				
STRNG	76	85#				
TAPEIN	219	227#				
TAPEO	171	179#				
TI1	228#	245				
TI2	229#	231	233			
TI3	234#	236	238			
TO1	182#	195				

CROSS REFERENCE COMPLETE



**Packaging  
Information**

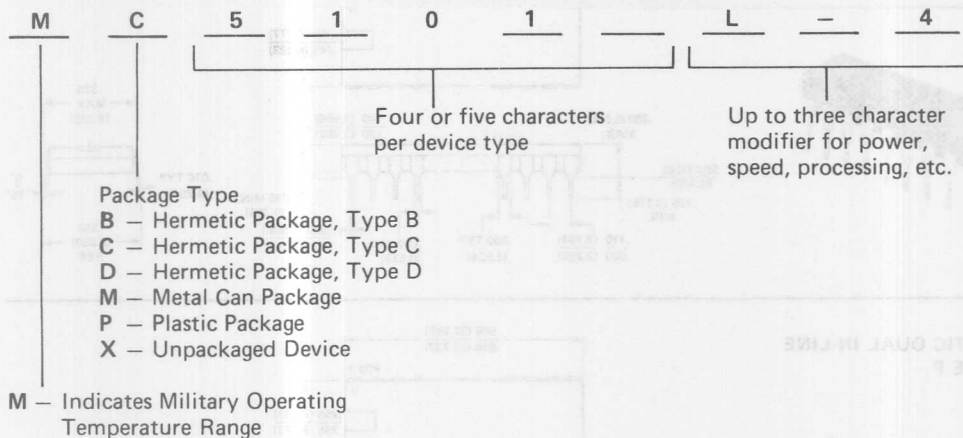
**MCS  
85**

**MCS  
85**

**MCS  
85**



## ORDERING INFORMATION



### Examples:

<b>P5101L</b>	CMOS 256 × 4 RAM, low power selection, plastic package, commercial temperature range.
<b>C8080A2</b>	8080A Microprocessor with 1.5 $\mu$ s cycle time, hermetic package Type C, commercial temperature range.
<b>MD3604/C</b>	512 × 8 PROM, hermetic package Type D, military temperature range, MIL-STD-883 Level C processing.*
<b>MC8080A/B</b>	8080A Microprocessor, hermetic package Type C, military temperature range, MIL-STD-883 Level B processing.*

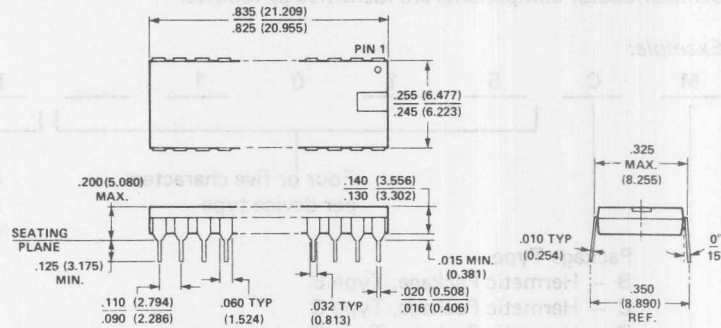
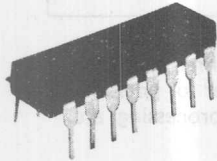
Kits, boards and systems may be ordered using the part number designations in this catalog.

The latest Intel OEM price book should be consulted for availability of various options. These may be obtained from your local Intel representative or by writing directly to Intel Corporation, 3065 Bowers Avenue, Santa Clara, California 95051.

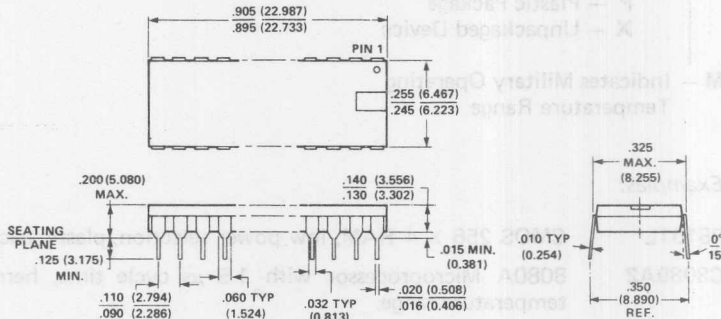
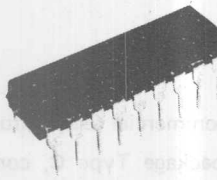
*\*On military temperature devices, B suffix indicates MIL-STD-883 Level B processing. Suffix C indicates MIL-STD-883 Level C processing. "S" number suffixes must be specified when entering any order for military temperature devices. All orders requesting source inspection will be rejected by Intel.*

**PLASTIC DUAL IN-LINE PACKAGE TYPE P**

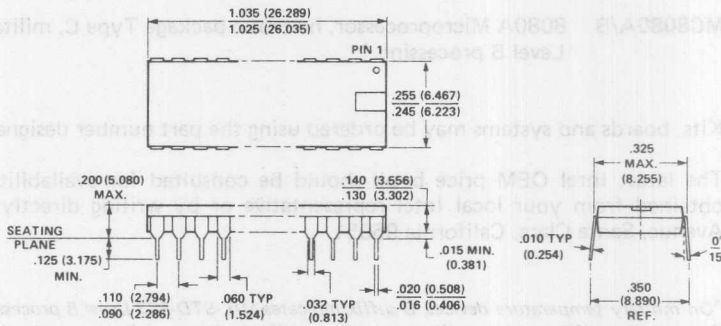
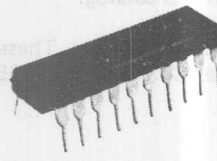
**16-LEAD PLASTIC DUAL IN-LINE  
PACKAGE TYPE P**



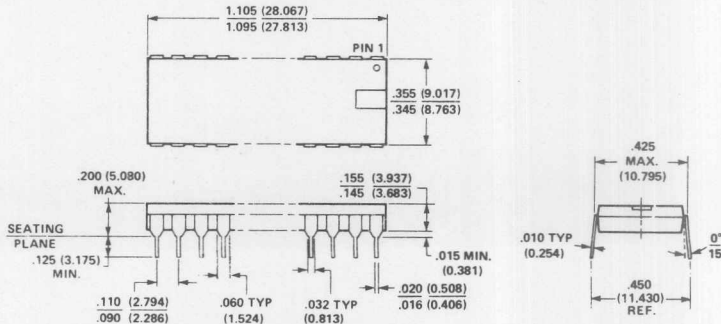
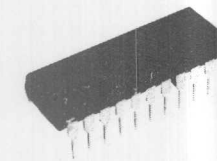
**18-LEAD PLASTIC DUAL IN-LINE  
PACKAGE TYPE P**



**20-LEAD PLASTIC DUAL IN-LINE  
PACKAGE TYPE P**



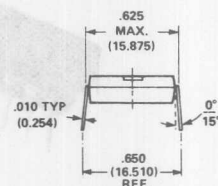
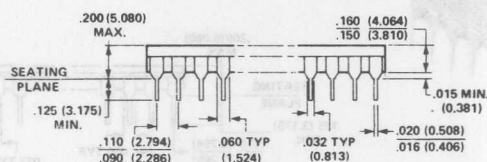
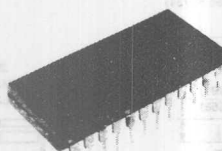
**22-LEAD PLASTIC DUAL IN-LINE  
PACKAGE TYPE P**



## PLASTIC DUAL IN-LINE PACKAGE TYPE P

Top view dimensions:

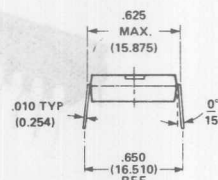
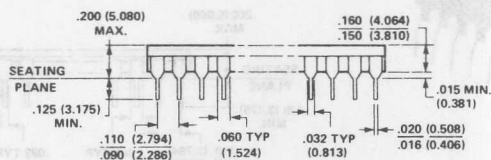
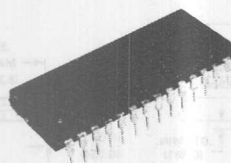
- Overall width: 1.245 (31.623)
- Width of top surface: 1.235 (31.369)
- Pin 1 location: PIN 1
- Distance from top surface to pin 1: .545 (13.843)
- Distance from bottom surface to pin 1: .535 (13.589)



1.455 (36.957)  
1.445 (36.703)

PIN 1

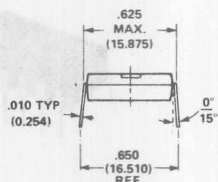
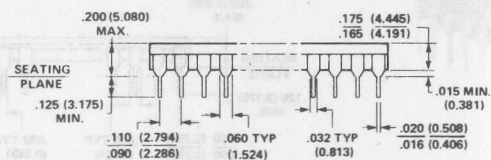
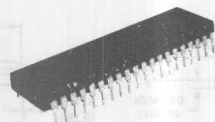
.545 (13.843)  
.535 (13.665)



2.055 (52.197)  
2.045 (51.943)

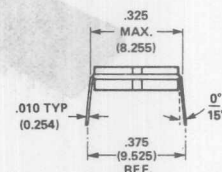
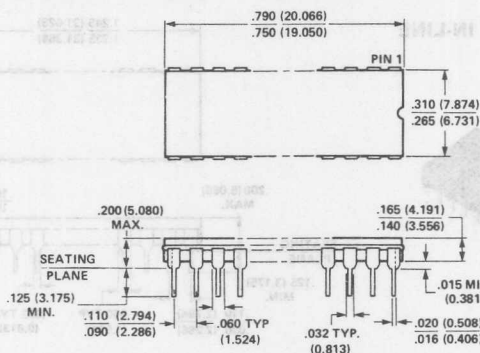
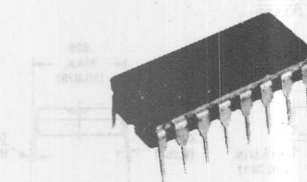
PIN 1

0.545 (13.843)  
0.535 (13.589)

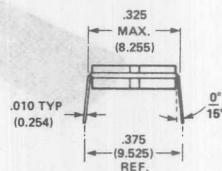
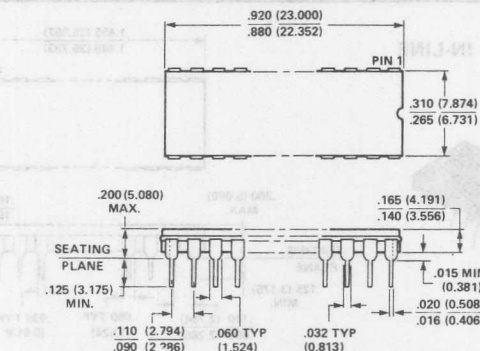
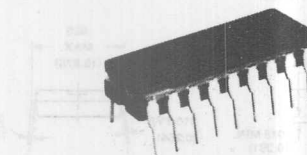


## CERAMIC DUAL IN-LINE PACKAGE TYPE D

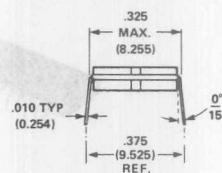
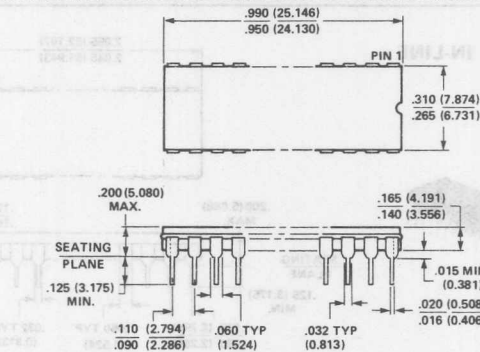
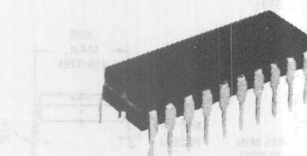
### 16-LEAD HERMETIC DUAL IN-LINE PACKAGE TYPE D



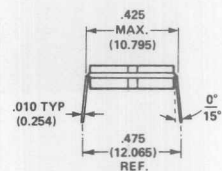
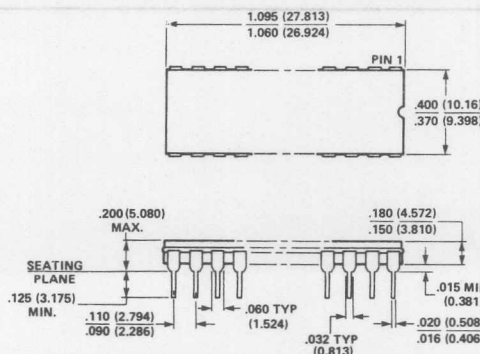
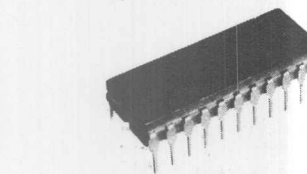
### 18-LEAD HERMETIC DUAL IN-LINE PACKAGE TYPE D



### 20-LEAD HERMETIC DUAL IN-LINE PACKAGE TYPE D



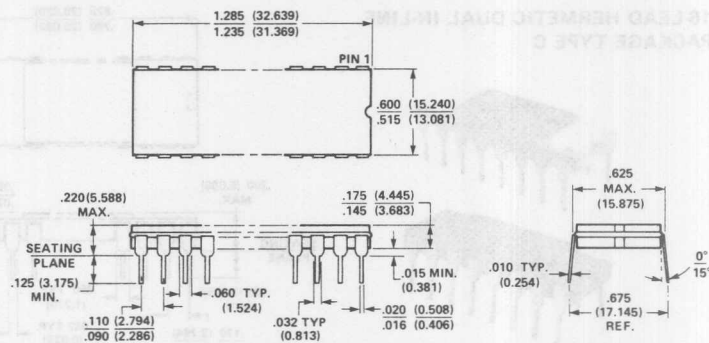
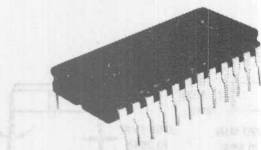
### 22-LEAD HERMETIC DUAL IN-LINE PACKAGE TYPE D



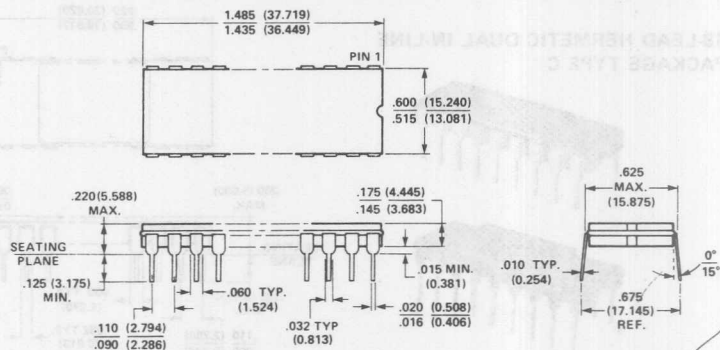
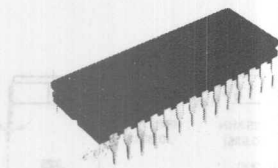
**PACKAGING INFORMATION** All dimensions in inches and (millimeters)

## CERAMIC DUAL IN-LINE PACKAGE TYPE D

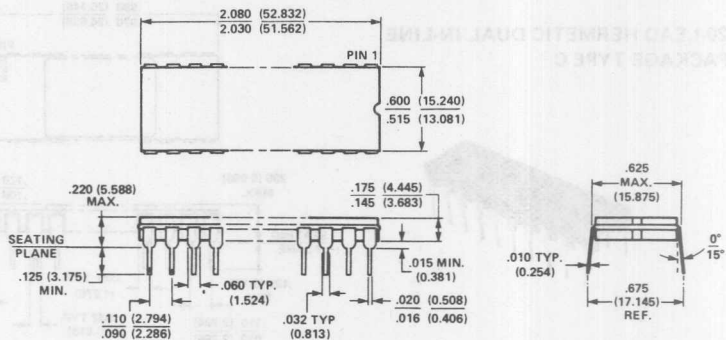
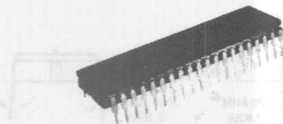
**24-LEAD HERMETIC DUAL IN-LINE  
PACKAGE TYPE D**



**28-LEAD HERMETIC DUAL IN-LINE  
PACKAGE TYPE D**

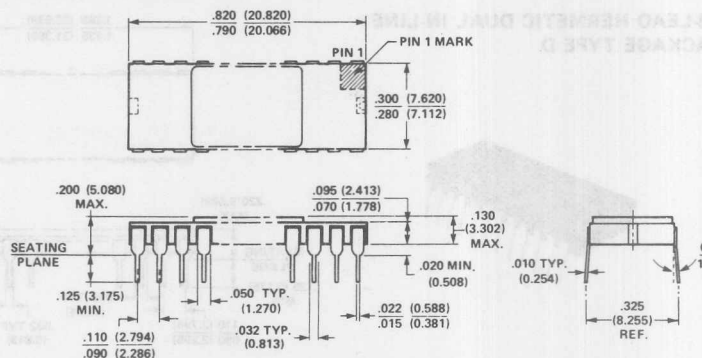
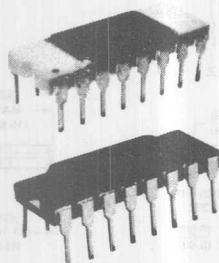


**40-LEAD HERMETIC DUAL IN-LINE PACKAGE TYPE D**

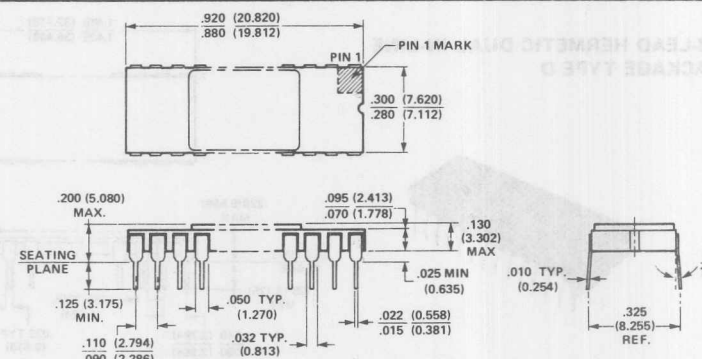
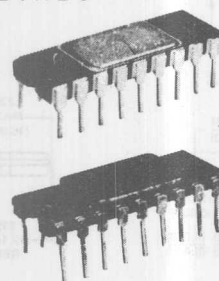


# **CERAMIC DUAL IN-LINE PACKAGE TYPE C**

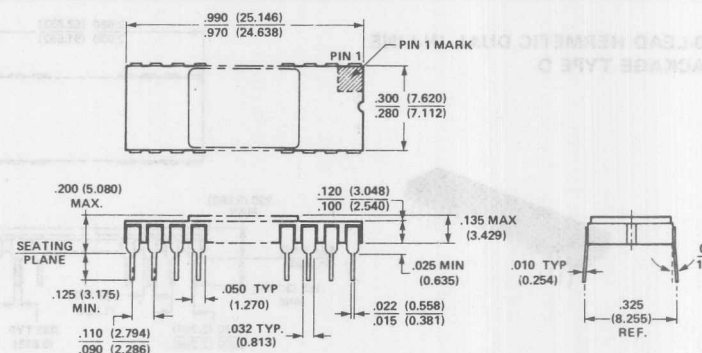
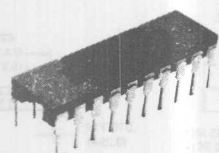
## **16-LEAD HERMETIC DUAL IN-LINE PACKAGE TYPE C**



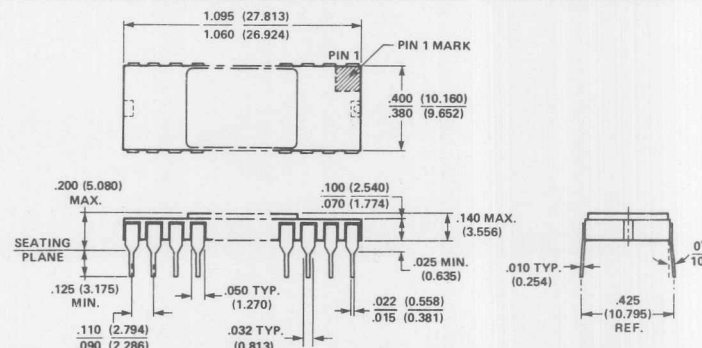
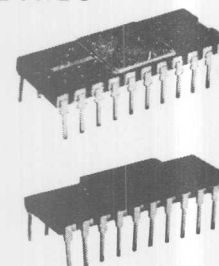
## **18-LEAD HERMETIC DUAL IN-LINE PACKAGE TYPE C**



## **20-LEAD HERMETIC DUAL IN-LINE PACKAGE TYPE C**



## **22-LEAD HERMETIC DUAL IN-LINE PACKAGE TYPE C**

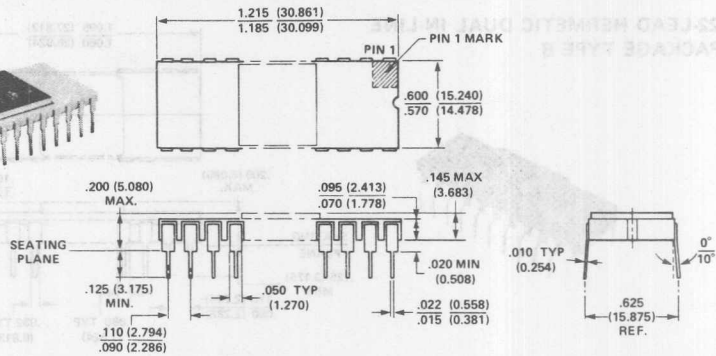
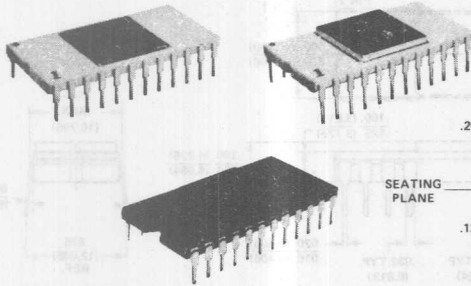


## PACKAGING INFORMATION

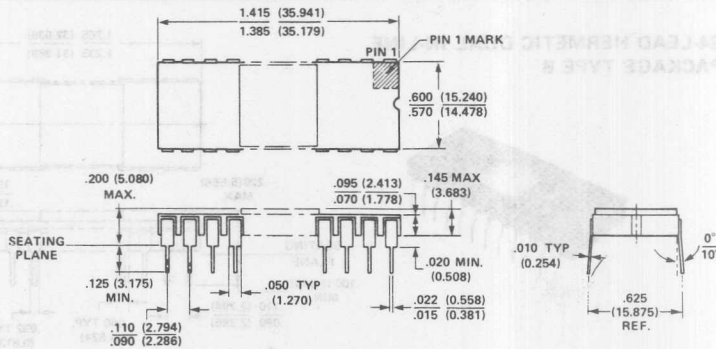
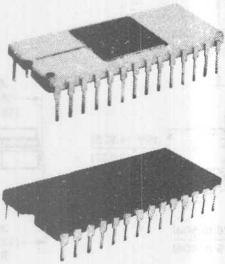
All dimensions in inches and (millimeters)

## CERAMIC DUAL IN-LINE PACKAGE TYPE C

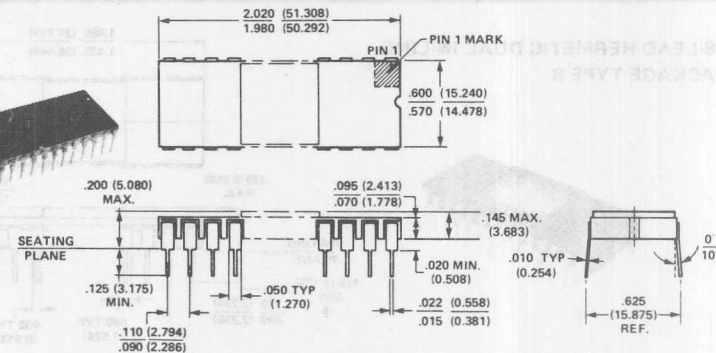
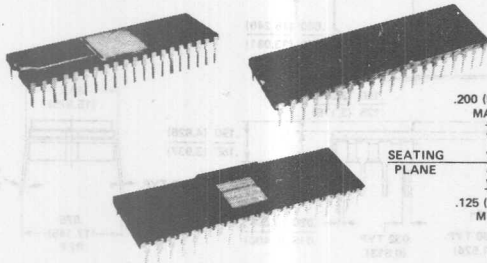
**24-LEAD HERMETIC DUAL IN-LINE  
PACKAGE TYPE C**



**28-LEAD HERMETIC DUAL IN-LINE  
PACKAGE TYPE C**

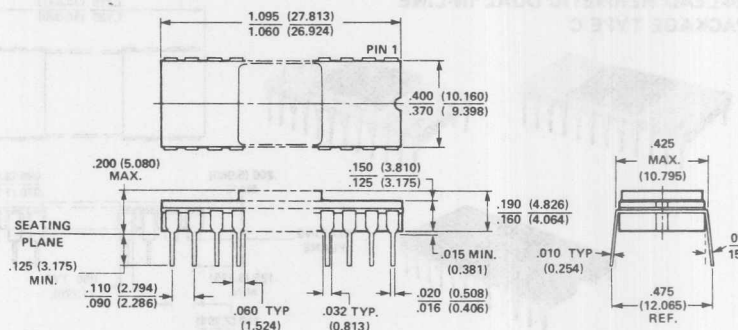
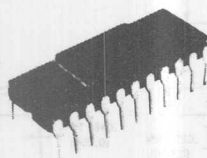


40-LEAD HERMETIC DUAL IN-LINE  
PACKAGE TYPE C

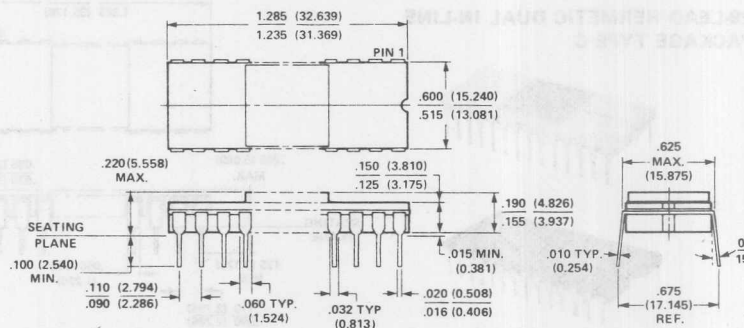
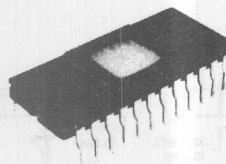


## CERAMIC DUAL IN-LINE PACKAGE TYPE B

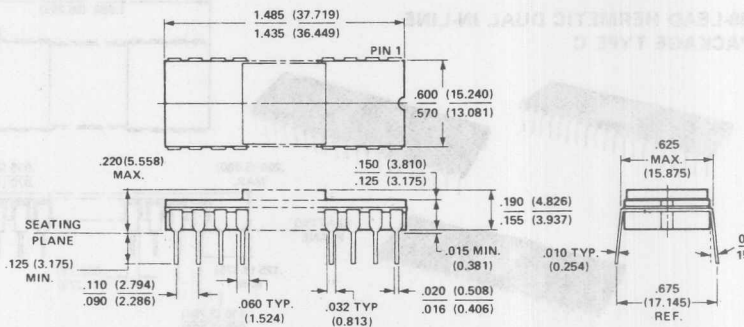
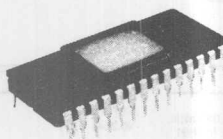
### 22-LEAD HERMETIC DUAL IN-LINE PACKAGE TYPE B



### 24-LEAD HERMETIC DUAL IN-LINE PACKAGE TYPE B



### 28-LEAD HERMETIC DUAL IN-LINE PACKAGE TYPE B



### 40-LEAD HERMETIC DUAL IN-LINE PACKAGE TYPE B

